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 4, Issue 3, February 2024

Air-Writing Recognition Using Convolution Neural Network

Tanvi Vilas Bodke[1], **Sampada Ramanth Shinde**[2], **Shravani Dashrath Lohote**[3], **Pavan Bhausaheb Shinde**[4]

Department of Information Technology[1,2,3,4]

Amrutvahini Polytechnic, Sangamner, India

Abstract*:* *Air based writing system refers to writing an alphabetical character or word in free space by moving a finger, marker, or handheld device. It is widely applicable where traditional pen-up and pen-down writing systems are troublesome. Due to the simple writing style, it has a great advantage over the gesture-based system. However, it is a challenging task because of the non-uniform characters and different writing styles. In this research, I developed an air-writing recognition system using three-dimensional (3D) trajectories collected by a depth camera that tracks the fingertip. For better feature selection, the nearest neighbour and root point translation was used to normalize the trajectory. We employed a convolutional neural network (CNN) as a recognizer. The model was tested and verified by the self-collected dataset. To evaluate the robustness of our model, we also employed the 2D convolutional neural network (2DCNN) alphanumeric character dataset and achieved best accuracy which is the highest to date. Hence, it verifies that the proposed model is invariant for digits and characters.*

Keywords: non-uniform characters ; air-writing recognition; convolutional neural network

I. INTRODUCTION

Writing in the air is a process to write something in a space by using gestures or trajectory information. It allows users to write in a touchless system. Especially, it is useful when traditional writing is difficult such as gesture-based interaction, augmented reality (AR), virtual reality (VR), etc. Air-writing solves such issues. In pen-and-paper-based systems, characters are written in a multi-stroke manner. However, air-writing has no such option, which represents its principal drawback. The gesture-based writing is a helpful way to avoid this problem, it is not an optimum solution due to the limitations of gesture varieties. Normally, the number of gestures is limited by human posture, but it is possible to increase the number of gestures by combining them. However, remembering all of these is a little difficult for new users. On the other hand, users can write in the same order as traditional methods in the air-writing system.

Air writing is performed in an imaginary box in front of a 2D or 3D camera. The principal problems when imposing a virtual boundary are spatiotemporal (related to space and time) variability and segmentation ambiguity. It is difficult for even the same user to repeat the same trajectory again in the same position and pattern. Recently, Leap Motion and Kinect cameras have greatly advanced the emerging field of 3D vision-based trajectory tracking; it is now possible to write in the air without the need for any extra device or sensor.

However, the deep learning-based approach obviates the need for manual feature generation. Therefore, we implemented deep learning-based algorithms, named the convolutional neural network (CNN).

The main contributions of this paper are as follows.

We design deep learning models allowing accurate air-writing digit recognition. The proposed CNN networks are robust under different experimental conditions such as normalized/non-normalized situations. However, the 2D CNN provides a better result, especially for a distorted case.

We create a large publicly available dataset that will play a vital role in deep learning and machine learning. The dataset contains 21,000 trajectories, which is sufficient for any deep learning algorithm. Among those, 1000 test trajectories are also included to measure the model’s performance in the unknown data.

DOI: 10.48175/568

II. LITERATURE SURVEY

Standalone devices have been used for air-writing and gesture-based digit recognition. they are readily applicable and serve as a prototype for user interface (UI) designers who lack knowledge of pattern recognition. Performance has been compared among various algorithms and devices. All current devices are handheld or attached to the body. Importantly, the sensors operate without external control. Researchers have focused on reducing the error from trajectory by manipulating the velocities and signals of inertial sensors .Many of them have been focused on designing efficient and effective algorithms to minimize the tradeoff between complexity and accuracy.

Nowadays CNN is also becoming popular for different applications for its different variants and robustness. For this research, we also used a depth-wise CNN network.

Recently, researchers have been following different strategies for air-writing recognition. Nguyen and C. Bartha [5] introduced shape writing and compared it with the Swype on a smartphone. It is shown that shape writing performs better as a virtual keyboard. Amma et al. [6] proposed an air-writing system using the inertial measurement unit, which required attaching with the hand. The HMM recognizer and SVM classifier were used to detect the character in the air. The main drawback of this work is that it is handheld; always attaching this device with the hand is very difficult and tedious. To overcome this situation, Qu et al. [1] proposed a Kinect based online digit recognition system using DTW and SVM. Lately, Mohammadi and Maleki proposed a Kinect based Persian number recognition system. However, those are not full writing systems. Chen et al. [4] and Kumar et al. [2] proposed a full writing system in the air including character and word. Both used Leap Motion as a trajectory detection device and HMM as a recognizer, but the accuracies were not significant. However, Kumar et al. [2] used the BLSTM algorithm and showed that the accuracy was higher than HMM. Most of the research has been done by using the trajectory information directly, i.e., using the temporal information. On the other hand, Setiawan and Pulungan [8] proposed a 2D mapping approach, in which trajectories were collected by the Leap Motion device and converted to the 2D image matrix like the popular MNIST dataset. Nowadays, Wi-Fi and Radar-based technology have become popular. Fu et al. [3] Proposed a Wi-Fi device-based (named as Wi-Fi) method using principal component analysis (PCA) and HMM algorithm. The main drawback of this work is that accuracy is not reasonable. However, Arsalan and Santra [7] solved the accuracy issues using millimeter-wave radar technology. They used three radars calibrated with trilateration techniques to detect and localize the hand marker, which is troublesome to implement for real-life applications. Therefore, we were motivated to develop a vision-based and hassle-free system for all users. At the same time, we achieved very good recognition accuracy.

III. METHODOLOGY

The whole process is divided into four principal parts fingertip detection, data collection, normalization, and network design. A complete block diagram is shown in Figure i to describe the details of the proposed method. The following subsections provide an in-depth explanation for each part.

3.1 Fingertip Detection

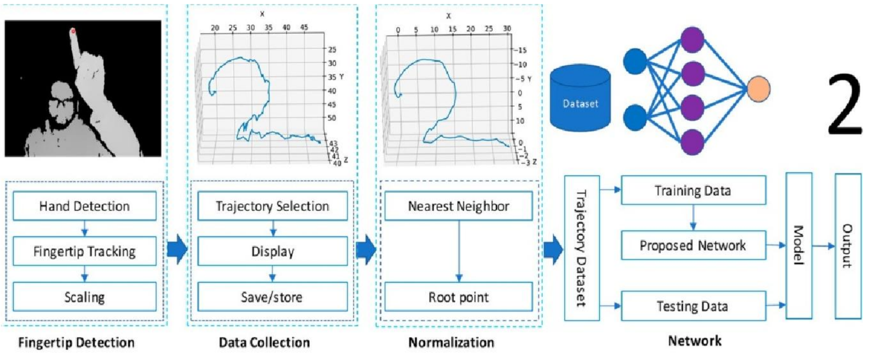


Figure i. Block diagram for the proposed method including the fingertip detection, data collection, normalization, and network design.

Fingertip detection was performed by an Intel RealSense SR300 camera. It is a widely used TOF (time-of-flight) camera for gesture detection, finger joint tracking, and depth-sensing research area. Firstly, hand segmentation and detection were done. Normally, there are 22 finger joints in our hands. Amongst them, the index fingertip was tracked for trajectory writing for user convenience. However, the trajectory was drawn through a virtual window. To fit and display in the physical window (such as a computer screen), scaling was required. We calculated the physical distance by multiplying the window size and the adaptive value for both x (horizontal) and y (vertical) directions. The adaptive value is the normalized value between 0 and 1 collected through the RealSense camera. In the User Interface (UI), the window sizes were 640 and 480 for x and y, respectively.

3.2 Data Collection

A simple interactive UI was designed requiring minimal instructions. The writing order is shown in Figure ii, which is similar to the traditional writing order. We did not apply graffiti or uni-stroke writing constraints. The stroke orders for each letter were defined in the usual manner i.e., the box-writing process. There are two types of approaches to recognize the air-writing system, online and offline. Here in, we followed an offline method which is also known as the 'push-to-write' approach to collect the data. In this process, users were requested to write digits in front of the depth camera, and that was collected as a spatial trajectory sequence.

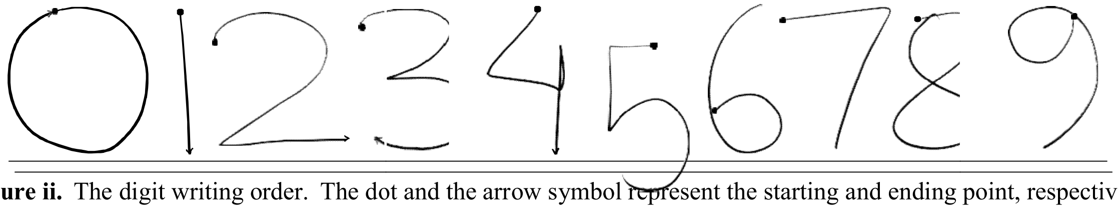


Figure ii. The digit writing order. The dot and the arrow symbol represent the starting and ending point, respectively.

3.3 Normalization

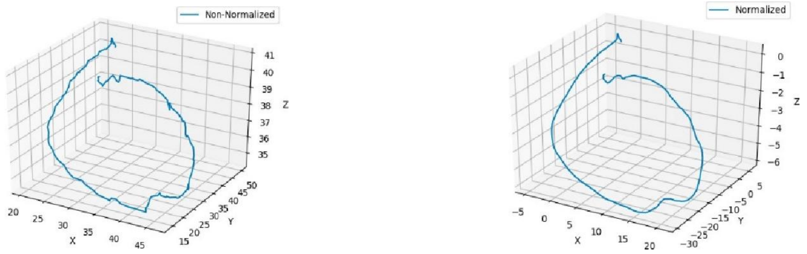
The main challenge for air-writing is that the trajectory is zigzag, i.e., not smooth, thus requiring normalization before feeding the network. We employed two normalization techniques—the nearest neighbor and root point. The details are as follows:

Nearest Neighbor Point Normalization:

The nearest neighbor point normalization technique is simple and heuristic. As it is not a pen-up/pen-down system, some displaced points are captured, which may change the trajectory shape. To deal with this situation, averaging the nearest-point transformation is used to change the deviated line to a smooth and straight line.

Root Point Normalization

During air-writing, users write in an imaginary ("virtual") box in the air. All digits are written in the first quadrant in a cartesian plane but in a different position. The virtual box does not have fix boundaries or margins; thus, the same digit may be written in different positions, even by the same user. This causes a random initial position. We used root point translation to generalize the starting point.



(**a**) Sample trajectory before normalization and (**b**) Trajectory after normalization

Figure iii. Three-dimensional trajectory visualization

The non-normalized digit 0 is shown in Figure iii(a); it is noisy and has a zigzag effect, while the fully normalized trajectory (Figure iii(b)) is smooth. In this figure, the *x* and *y* are the distance value in the virtual window; i.e., the passing distance between the start and end position. The *z* is the distance between the hand fingertip and the camera. All are in centimeters (cm). The negative value in Figure iii(b) indicates the relative distance.

In this research, we employed two state-of-the-art neural network algorithms, CNN. CNN work based on convolution and recurrent units, respectively. Both are widely applicable for time series prediction.

CNN Network

A CNN network consists of an input and output layers composed of multiple hidden layers. Hidden layers include convolution, pooling, and activation layers. CNN has different variants based on the application and dataset. The normal CNN network contains lots of calculations due to the convolution and pooling layer. Hence, we applied the separable convolution layer which is faster than the general CNN layer. It is widely known as a depth wise convolution.

The proposed CNN network is shown in Figure iv. Like the LSTM network, the input layer is also set to the maximum length of the trajectories. The input layer is a 300 × 1-dimensional vector that transfers the input value to the 1st separable convolution layer. Convolution layers 1, 2, and 3 contain 64, 128, and 256 channels, respectively. In all the cases the filter size is 3. Each convolution

Layer is associated with the 1D max-pooling layer. There are two dense layers containing 256 and 128 neurons, respectively. Dropout rate 0.5 is used in the DENSE1 and DENSE2 layers. The ReLU Equation (14) activation function is used for all the cases except the output layer. The softmax Equation

(15) Regularization is used in the output layer. Adam is used as an optimizer with a learning rate of 0.0001 and categorical cross-entropy as a loss function.

3.4 Network Design

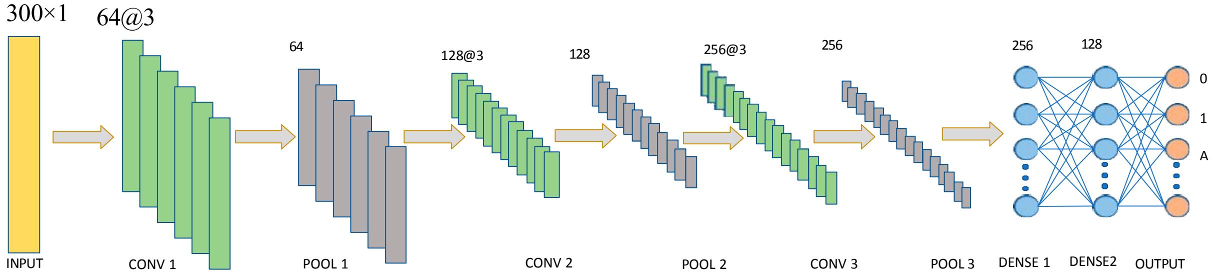
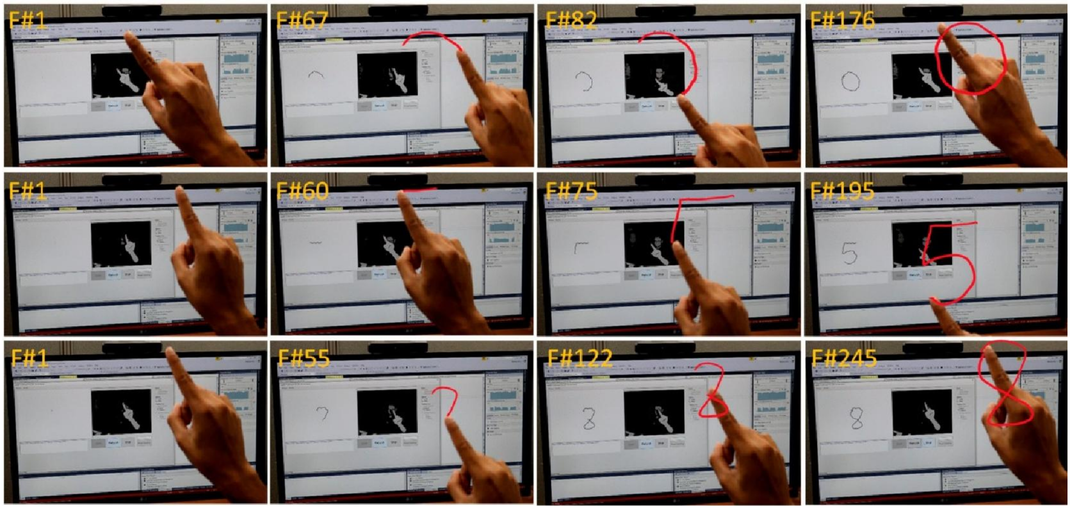


Figure iv. Proposed convolutional neural network (CNN) model. The model has three convolutions, three max-pooling and 2 dense layers.

IV. EXPERIMENTAL SETUP

The experiment was done by connecting an Intel RealSense SR300 camera with a computer. A graphical user interface (GUI) was designed to collect the appropriate trajectory data. The C# and Python programming languages were used for interfacing and training, respectively. The proposed networks were implemented in Keras high-level API over the TensorFlow backend.

The trajectory was captured in real-time (50 fps). We used the NVIDIA GeForce GTX 1050 Ti graphics processing (GPU) unit with 32 GB memory to speed up the training process. The experimental environment is shown in Figure 5. The digit 0 was written in the air and tracked by the RealSense camera. It is not normalized; hence, the trajectory is a little bit distorted. Some of the complex digits and corresponding sample frames are shown in Figure 5. The frame by frame representation helps to understand the motion for the digit writing order. The trajectory capturing process started from the first frame (F#1). However, the ending frame is different for each individual character due to the different motion and writing patterns.



Experimental setup for air-writing dataset collection interface.

4.1. User Interface

In the user interface, there were three basic buttons to start, stop, and save trajectory. The GUI displays both depth information and the trajectory. The depth and the trajectory part were used to display the full hand indicating the tracked fingertip point and the captured trajectory, respectively. The GUI was very simple, so minimal instruction was needed. It was also highly interactive, featuring real-time trajectory capture and display. For simplicity, each digit was shown in 2D cartesian coordinates, but it is three dimensional. In Figure 5, digit 0 is shown on the left side; the depth maps of the finger and body appear on the right. As soon as the 'start' button clicked, the camera became active and started to detect the fingertip position. If the drawn trajectory is the expected trajectory, the user clicks the 'stop' button. If the trajectory is not expected or the user cannot draw properly, then they can click the 'Refresh' button to clear the window. Finally, the trajectory can be saved by pressing the 'Save' button. Users do not need to write down the label, the system itself incrementally generates one by one. Basically, the 'start' and 'stop' buttons in the UI control the initial and end points, respectively. To trim the unwanted starting or ending points, we used the 'cut' button to eliminate the terminal point according to the instructions.

V. CONCLUSION

In this paper, we proposed a air writing recognition system using an Intel RealSense SR300 camera and developed deep learning-based algorithms to recognize the trajectory. This is a paperless writing system. Researchers previously used different motion sensors and handheld devices; instead, we used a vision-based approach for better user experience. To verify the method and assess its accuracy, we proposed neural network models—CNN.

REFERENCES

[1] Qu, C.; Zhang, D.; Tian, J. Online Kinect Handwritten Digit Recognition Based on Dynamic Time Warping and Support Vector Machine. J. Inf. Comput. Sci. 2015, 12, 413–422.

[2] Kumar, P.; Saini, R.; Roy, P.P.; Dogra, D.P. Study of Text Segmentation and Recognition Using Leap Motion Sensor. IEEE Sens. J. 2017, 17, 1293–1301. [CrossRef]

[3] Fu, Z.; Xu, J.; Zhu, Z.; Liu, A.X.; Sun, X. Writing in the Air with WiFi Signals for Virtual Reality Devices. IEEE Trans. Mob. Comput. 2019, 18, 473–484.

[4] Chen, M.; AlRegib, G.; Juang, B.H. Air-Writing Recognition—Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data. IEEE Trans. Human-Machine Syst. 2016, 46, 436–444.

[5] Nguyen, H.; Bartha, M.C. Shape writing on tablets: Better performance or better experience? In Proceedings of the Human Factors and Ergonomics Society, Boston, MA, USA, 22–26 October 2012; pp. 1591–1593.

[6] Amma, C.; Georgi, M.; Schultz, T. Airwriting: A wearable handwriting recognition system. Pers. Ubiquitous Comput. 2014, 18, 191–203. [CrossRef]

[7] Arsalan, M.; Santra, A. Character Recognition in Air-Writing Based on Network of Radars For Human-Machine Interface. IEEE Sens. J. 2019, 19, 8855–8864. [CrossRef]

[8] Setiawan, A.; Pulungan, R. Deep Belief Networks for Recognizing Handwriting Captured by Leap Motion Controller. Int. J. Electr. Comput. Eng. 2018, 8, 4693–4704.

DOI: 10.48175/568