Trajectory-based Air-writing Character Recognition Using Convolutional Neural Network

Md. Shahinur Alam[1], Ki-Chul Kwon, and Nam Kim[2*]
Dept. of Computer and Communication Engineering
Chungbuk National University
Cheongju, Chungbuk, South Korea.
[1]research@shahinur.com, [2]namkim@chungbuk.ac.kr
*corresponding author

Abstract— **Writing in the air can be defined as to write digit or character in a 3D space by using a finger or marker movement. It is different from the traditional writing style. However, it has become easier to track finger and joint precisely due to the extensive improvement of sensor technologies. In this research, we proposed a trajectory-based air-writing character recognition system using a convolutional neural network (CNN). The trajectories were collected using a depth camera as a three-dimensional (3D) sequence. We used 10-fold cross-validation to validate the model. The accuracy of the proposed model was 97.29%. Also, we have collected an air-writing dataset containing 26,000 characters for training and validation, and another 4,000 for testing the model. The recognition time was 14ms per character which is fast enough to implement in a real-time system.**

Keywords-air-writing; CNN; character recognition; gesture recognition; human-computer interaction.

I. Introduction

Air-writing is a new form of user interfaces that enables us to write through gesture or finger movement in the air. Due to the different writing styles, it became more complicated than gesture recognition. The written characters are rendered on an imaginary plane without haptic or visual feedback. The user can only use a virtual coordinate to draw the character. The main problem is when it is written as a continuous motion trajectory, where each and every point is tracked. Air-writing is mostly suitable for pen and paperless interactions. It became possible due to the advancement of recent gesture and hand tracking devices. Basically, there are two types of approaches– wearable and non-wearable. Non-wearable devices are handheld requiring extra devices attaching with body or hand. On the other hand, non-wearable devices are mostly vision based, no need for extra devices or attaches something with the body. Hence, non-wearable devices are user-friendly and easy to use.

Gesture-based writing systems has been widely studied in the past. The limitations of the gesture-based writing system are that the number of gestures is limited by human posture. Therefore, we employed a trajectory-based writing system in the air. It enables the flexibility to the user not to bound in any special constraints. Users are free to write just in front of a camera considering a virtual window. The most challenging task is to track the hand motion and find out the writing intervals. Hence, researchers focused on different methods like online and offline handwriting recognition. The real-time recognition is called the online method. On the other hand, offline is not real-time and easier than online recognition. The data acquisition of the offline recognition is followed by the 'push to write' concepts [1] whereas the online recognition has no such things [2].

Accelerometer and gyroscope-based wearable devices were proposed for motion tracking [3]–[5]. The motion was wirelessly captured by using accelerometer and gyroscopes. Support vector machine (SVM) and HMM were employed for spotting and recognition for air-writing, respectively. A MAG-μIMU was proposed to develop a real-time estimation of hand motion [3]. An extended Kalman filter was utilized the gyroscope propagation for sensors, such as gravity, magnetometers or star trackers. A triaxial accelerometer-based digital pen was proposed to collect handwritten data [4]. The acceleration of hand motions was transmitted to the computer. The linear discriminator reduces the feature dimension, and these features were trained by the probabilistic neural network for recognition.

Kinect[6], [7] and Leap Motion [8]–[11] are the popular non-wearable device to track the hand motion in the air. Kinect is the long-range time of flight (TOF) camera whereas Leap Motion is the small range device. Qu *et al.* [6] proposed an online handwritten digit recognition system using Kinect depth and color information. Dynamic time warping (DTW) and support vector machine (SVM) was used to recognize the digit. Satiawan and Pulungan [12] proposed a Leap Motion controller based handwriting recognition system. But to get the essence of a full writing system, Kumar *et al.* [13] proposed a text segmentation and recognition system from word and text. Unlike the gesture and character recognition [1], Chen *et al.* [2] proposed a word recognition method to employ a full writing system. The word-based recognition rate was higher than the letter-based recognition. A combined finger spelling and the sign language recognition system was proposed to study the feasibility in different environment and methods [14]. This system can work in real-time. However, in this research, we used a mid-range Intel RealSense camera.

The HMM [1], [2], [13], [15] is a well-known machine learning algorithm for air-writing recognition. It is a statistical Markov model which contains hidden states. It can be considered as the simplest dynamic Bayesian network. The most important thing is that it can work well in a small dataset.

978-1-7281-4620-1/19/$31.00 ©2019 IEEE
DOI 10.1109/CRC.2019.00026

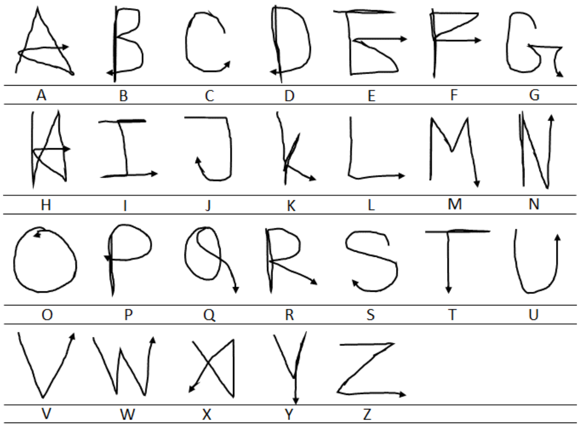


Figure 1. Character writing order.

Recently, long short-term (LSTM) and bi-directional LSTM (BLSTM) based recognition method has been proposed [13], [16]; but these algorithms required huge data. In this paper, we have proposed a convolutional neural network (CNN) to recognize the character. The main contributions of this paper are as follows:

1) We publish a large character dataset. To feed the deep learning algorithm, it requires huge data. In our dataset, there are 30,000 characters, among them, 26,000 are for training and validation. Rest of the data are for testing and verify the model.
2) We propose a simple and robust CNN model for air-writing recognition. The accuracy verifies the model performance.

The rest of the paper is organized as follows. We discuss the dataset in section II. The proposed CNN network is described in section III. The experimental setup and results are described in section IV and V, respectively. Section VI contains the future plan and conclusion.

II. The Dataset

In this section, we discuss the dataset, the data collection procedure, and its parameters.

A. Data Collection

The character was collected as a sequence of trajectory. The fingertip was considered as an alternative of a pen in the traditional cursive writing style. An Intel RealSense SR300 camera was used to track the trajectory. An interactive user interface was designed to make the system easier. Users can write a character in front of a camera and this camera detects the fingertip and collect those fingertips as a sequence as follows-

$$A= \{x_1, y_1, z_{1,} x_2, y_2, z_2 \ldots\ldots x_n, y_n, z_n\} \qquad (1)$$

where, A is the labeled character; x, y, z represents the x-axis, y-axis, and z-axis, respectively; where 1 is the first point and n is the last point of a trajectory. The length can vary based on the users writing style and speed.

TABLE 1. Data Parameters

Character	*Maximum*	*Minimum*	*Mean*	*Variance*	*STD*
A	156	40	88	398	19
B	141	46	84	189	13
C	96	21	37	80	9
D	136	46	70	122	11
E	151	56	89	194	13
F	169	50	87	251	15
G	155	41	75	210	14
H	167	47	94	257	16
I	155	53	87	236	15
J	121	33	54	104	11
K	144	44	75	183	13
L	122	25	43	86	9
M	173	45	89	177	13
N	127	44	77	145	12
O	82	25	46	52	7
P	131	41	65	113	10
Q	130	38	64	125	11
R	144	48	77	167	12
S	95	26	45	76	8
T	118	32	57	117	11
U	83	24	41	45	6
V	107	36	45	67	8
W	122	42	71	111	11
X	126	39	61	130	11
Y	108	31	60	95	10
Z	130	35	56	105	10

The writing direction is shown in Fig. 1. It is a little different from the traditional multistroke writing style; and written in an unistroke manner [17]. For example, the character H is written in unistroke style, it doesn't look like the exact character we usually see in the book or any printed media. This is the main problem to differentiate similar characters like F, R, and U, V.

B. Data Parameters

Our RealSense trajectory character (RTC) dataset contains 30,000 trajectories[1]. Amongst them, 26,000 is for training and validation; and 4,000 is for testing the model accuracy. Table.

[1] Dataset available: https://www.shahinur.com/en/RTC

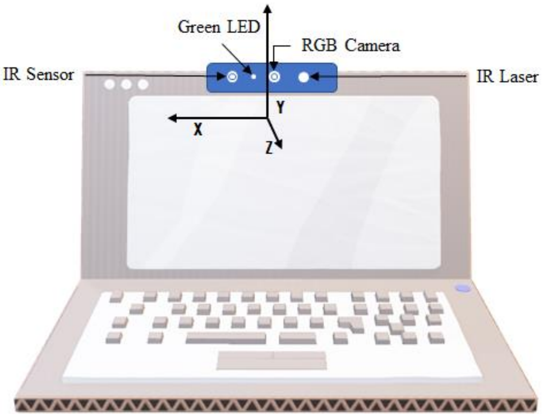


Figure 2. Coordinate system.

1 shows the detailed parameters of the RTC dataset. The trajectories are not the same and vary with length. The maximum and minimum length of the trajectory of an individual character determines the highest and lowest sequence. The mean (1) value determines the average length. Since the difference between maximum and minimum length is high, it is difficult to get the in-depth information about the data. On the other hand, variance (2) and standard deviation (STD) provides substantial information. A little deviation of STD (3) indicates that the length of the most trajectories varies a little. It is observed from the Table. 1 that the longest character is H and the smallest character is O.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

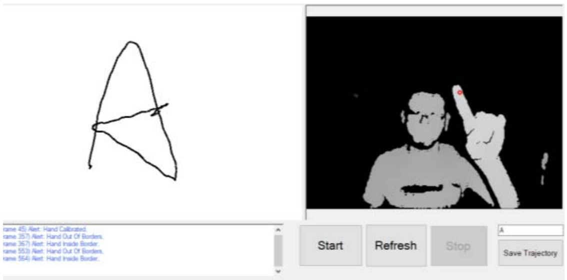
$$\mathrm{S}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{2}$$

$$\mathrm{S} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3}$$

where $\bar{\mathrm{x}}$ is the mean value, x_i is the i^{th} datum, S^2 is the variance and S is the STD. The variance helps us to define the input length of the network.

C. The Coordinates

The coordinates are the same as the camera coordinates. In Fig. 2, the basic camera components are shown. It has RGB and IR sensor. IR sensor is used to detect the depth. The vertical and horizontal direction from the camera is the x-axis and y-axis, respectively. The distance between the fingertip to the camera is the depth and this represents the z-axis. The depth range of this camera is 0.2 to 1.5 meter. We captured all character in a range between 0.5 to 1 meter.



rame 45) Alert: Hand Calibrated.
rame 357) Alert: Hand Out Of Borders.
rame 367) Alert: Hand Inside Border.
rame 553) Alert: Hand Out Of Borders.
rame 564) Alert: Hand Inside Border.

Figure 3. Graphical user interface

III. NETWORK DESIGN

The proposed network consists of input, convolution, dense and output layers. It is shown in Fig. 4 that there are 2 convolution, 2 pooling, and 2 dense layers. The convolutional kernel size is 3×3. The filter size for Conv1 and Conv2 was 64 and 128, respectively. We used ReLu (4) as an activation function in both convolution (Conv1 and Conv2) and dense (Dense1 and Dense2) layer. ReLu is commonly used activation function which simply eliminates the negative term and keeps the positive term. The max pooling is used in the pooling1 and pooling2 layer to down sample the most important feature. The softmax (5) activation function is used in the output layer.

$$\mathrm{f}(x) = \max(0, x) \tag{4}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}} \tag{5}$$

This function converts the output as a vector, based on the probability distribution. To prevent the overfitting, dropout [18] is used in the Conv1 and Conv2 layer. Dropout is a regularization technique, where neurons are randomly selected and ignored during training. This results in a network that becomes capable of being less likely to overfit training data. We have employed categorical cross-entropy to validate the

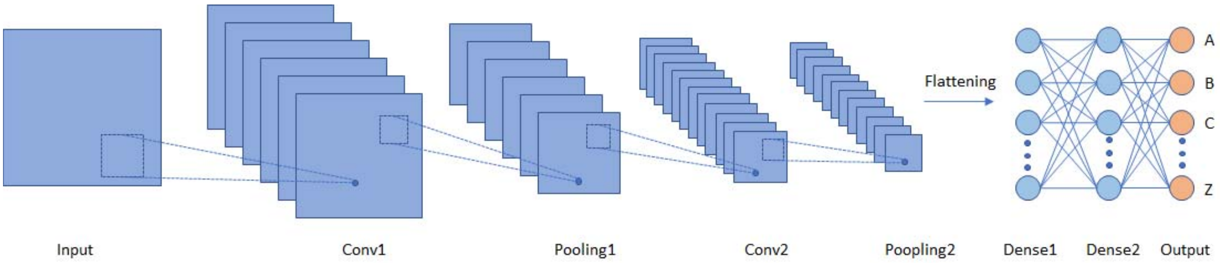
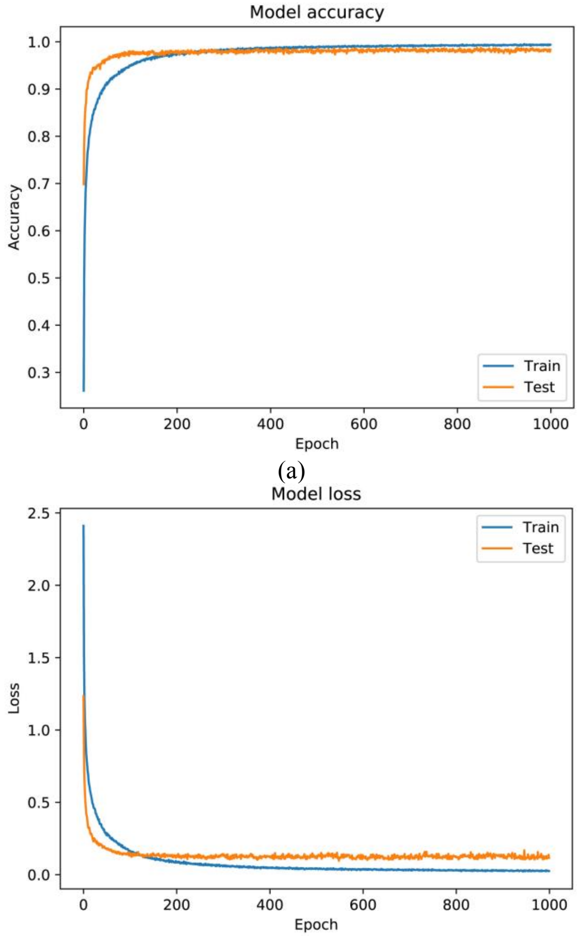


Figure 4. Proposed CNN Model



(a)

(b)

Figure 5. Model loss and accuracy: (a) accuracy over iterations, (b) loss over iterations.

model accuracy. It measures the performance of the model whose output is between 0 and 1 according to the probability distribution. Adam [19] is used as an optimizer with learning rate 0.0001. It is an adaptive learning rate optimizer that is designed for training a deep neural network.

IV. Experimental Setup

The experimental setup was easy and simple. An Intel RealSense SR300 camera was employed with a computer. The system runs in 16 GB memory, Intel Core i5 processor and GeForce GTX 1050 Ti GPU. The C# and Python programming language were used to design the user interface and to build the model, respectively. The network was designed with the help of Keras over TensorFlow backend.

An interactive user interface was designed (shown in Fig. 3) to collect the trajectory information. There are two windows, one is for displaying the acquired trajectory and another is for visualizing the raw depth information. Users sat down in front of the camera and drawn the character in the air. They can start, stop and capture using specified buttons. The trajectories were collected at 50 frames per second.

V. Results and Discussions

The overall accuracy of our proposed method was 97.29%. However, the recognition accuracy for most of the characters was good. But the recognition accuracy is not uniform for all the characters. We got the worst recognition accuracy for character ‘F’, because of its writing style. This character is similar to the ‘E’ and ‘R’ (Fig. 1). From Table. 1, we can see that the length of those 3 characters is also similar. It is observed that the false positive rate is higher when the shape and the length are very much similar to each other.

The model was trained by the 26,000-character using 10-fold cross-validation, and the accuracy measured by the other 4,000 data. It is observed from the model loss and accuracy that it became stable after 300 iterations. Fig. 5(a) and Fig. 5(b) shows the accuracy over iteration and loss over iteration, respectively. We measured the loss function using categorical cross-entropy. Both cases, the graph is consistent and smooth, which is expected to train the model. The testing time was 14 milliseconds for each individual character. Due to the fast recognition time, it is applicable in real-time systems.

We conducted a comparative analysis between this proposed method and the previous work. There is some substantial work that has already been done. In the earlier method of air-writing, accelerometers and gyroscopes based devices were used to track the motion [5]. Their HMM-based classifier achieved 83% recognition accuracy. Later, a controlled trajectory-based method was proposed [20]. They studied both user-dependent (UD) and user-independent (UID) cases. The accuracy was 96.2% and 91.2% for UD and UID, respectively. The leap motion based method was proposed by Kumar et al. [13]. Their accuracy was 86.88% and 81.25% for BLSTM and HMM, respectively. Recently, the WiFi signal based air-writing system has been proposed [15]. They employed two methods- PCA and HMM. The accuracy is higher for HMM-based classifier. The overall accuracy for character and word is 94% and 88.74%, respectively. Compared to others, our CNN based method shows relatively good accuracy. Though we considered only the character recognition system, we hope that this dataset will help to recognize words by analyzing the connecting ligature.

VI. Conclusion

In this work, we collected an air-writing character dataset using Intel RealSense SR300 camera containing 30,000 trajectory data. A CNN model was also proposed to recognize the character. To split the data for validation purpose, 10-fold cross-validation was used. Categorical cross-entropy was employed to measure the model loss, based on the accuracy.

The experimental result of training and testing was reasonable, but we think more improvement is possible. In our future work, we will try to improve the performance and accuracy. Also, we investigated that some character has a higher error rate, we will try to fix it using some special feature processing technique in the future.

Acknowledgment

This research was supported by the Korea Government, under the ITRC (Information Technology Research Center) support program (IITP-2019-2015-0-00448) supervised by the IITP (Institute for Information & communications Technology Promotion) and was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. NRF-2017R1A2B4012096 & No. NRF-2018R1D1A3B07044041).

References

[1] M. Chen, G. AlRegib, and B.-H. Juang, "Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting Motions," *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 3, pp. 403–413, Jun. 2016.

[2] M. Chen, G. AlRegib, and B. H. Juang, "Air-Writing Recognition - Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data," *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 3, pp. 436–444, 2016.

[3] Y. Luo *et al.*, "An Attitude Compensation Technique for a MEMS Motion Sensor Based Digital Writing Instrument," in *2006 1st IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, 2006, pp. 909–914.

[4] J. S. Wang and F. C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, 2012.

[5] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," in *Personal and Ubiquitous Computing*, 2014.

[6] C. Qu, "Online Kinect Handwritten Digit Recognition Based on Dynamic Time Warping and Support Vector Machine," *J. Inf. Comput. Sci.*, vol. 12, no. 1, pp. 413–422, 2015.

[7] S. Poularakis and I. Katsavounidis, "Low-Complexity Hand Gesture Recognition System for Continuous Streams of Digits and Letters," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2094–2108, 2016.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 162–175, Jan. 2019.

[9] C. Palmero, J. Selva, M. A. Bagheri, and S. Escalera, "Recurrent CNN for 3D Gaze Estimation using Appearance and Shape Cues," May 2018.

[10] H. Ranganathan, S. Chakraborty, and S. Panchanathan, "Multimodal emotion recognition using deep learning architectures," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.

[11] J. K. Sharma, R. Gupta, and V. K. Pathak, "Numeral Gesture Recognition Using Leap Motion Sensor," in *Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015*, 2016.

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

[13] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, "Study of Text Segmentation and Recognition Using Leap Motion Sensor," *IEEE Sens. J.*, 2017.

[14] P. Kumar, R. Saini, S. K. Behera, D. P. Dogra, and P. P. Roy, "Real-time recognition of sign language gestures and air-writing using leap motion," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, 2017, pp. 157–160.

[15] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the Air with WiFi Signals for Virtual Reality Devices," *IEEE Trans. Mob. Comput.*, vol. 18, no. 2, pp. 473–484, Feb. 2019.

[16] S. Xu and Y. Xue, "A Long Term Memory Recognition Framework on Multi-Complexity Motion Gestures," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 201–205, 2018.

[17] S. J. Castellucci and I. S. MacKenzie, "Graffiti vs. unistrokes," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, 2008, p. 305.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014.

[20] Y. Zhou, Z. Dai, and L. Jing, "A controlled experiment between two methods on ten-digits air writing," *Proc. - 2016 16th IEEE Int. Conf. Comput. Inf. Technol. CIT 2016, 2016 6th Int. Symp. Cloud Serv. Comput. IEEE SC2 2016 2016 Int. Symp. Secur. Priv. Soc. Netwo*, pp. 299–302, 2017.