The Visual Computer (2022) 38:1015–1025
https://doi.org/10.1007/s00371-021-02065-x

ORIGINAL ARTICLE

On-air English Capital Alphabet (ECA) recognition using depth information

Hasan Mahmud[1] · Robiul Islam[2] · Md. Kamrul Hasan[1]

Accepted: 6 January 2021 / Published online: 30 January 2021

Abstract

On-air writing can be considered as a time-dependent event where hand gesture is produced in a natural environment through index finger movement. A sequence of such movements containing several time steps in 3D space can be utilized to construct an English Capital Alphabet (ECA). While Previous researches investigated 2D features, we believe that depth information may play a significant role along with other features in recognition of these dynamic gestures. We have captured hand finger motion information using a depth camera and represented them as depth images for each ECA. The hand finger trajectory data were extracted from the depth image, and a combination of depth-based features and non-depth features were generated; depth variation was performed in the depth-based features, and then, all the feature values were converted into time-series data. Dynamic Time Warping distances were determined between a template ECA and a test ECA for each ECA collected from 15 participants. These distance-based features were then fed into a multi-class SVM for training and testing and got the recognition accuracy of 80.77% without depth and 88.21% with depth-based features. To cope with the over-fitting problem, we applied the resampling technique and got the highest recognition accuracy of 96.85%, and at last, we applied some feature selection techniques to analyze the recognition results.

Keywords Air-writing · Gesture recognition · Depth information · Time-series data · Dynamic time warping · Support vector machine

1 Introduction

Air-writing can be defined as a motion-oriented activity of hand or finger in the free space to represent a linguistic character. The idea of recognizing 'air-writing' was incubated by Amma [2] where he tried to recognize ECAs using wired device. The recognition of on-air alphabet writing is a part of broader gesture recognition research [19], kind of dynamic gesture recognition and air-writing might seem to be similar to online handwriting recognition [13] task. In this process, a user can lift his/her hand from the touchpad. However, in air-writing, it is difficult to differentiate which movements are part of writing and which movements are not. Consequently, many different extra strokes are mixed up with the actual writing complicating the recognition process. Moreover, while writing in the air, the hand may be near the face or the body and their similar color might be confusing due to occlusion. To overcome this problem, many researchers have used special markers [2] around the writing finger. A special version of air-writing can be to write on a surface (which is not touchpad), because people feel natural writing on a surface.

The use of depth information (user distance from the camera) provided by depth camera (e.g., Microsoft Kinect, Intel Real Sense) helps to segment the hand where the traditional cameras will fail. Thanks to the depth camera for making hand segmentation and tracking process easier and faster without ambiguity. The depth information helps to generate depth image and used as skeleton features to different gesture recognition systems [16]. More importantly, this depth

✉ Hasan Mahmud
hasan@iut-dhaka.edu

Robiul Islam
rislam@edu.hse.ru

Md. Kamrul Hasan
hasank@iut-dhaka.edu

[1] Systems and Software Lab, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh

[2] Laboratory of Computational Pragmatics Models and Methods, Higher School of Economic, Pokrovsky Boulevard, 11, Moscow, Russia

information can be effectively utilized to represent depth-based features along with non-depth features. On-air writing requires index finger to move not only left/right (X-axis) or up/down (Y-axis) but also forward/backward (Z-axis). In the free space, a user is not using any 2D surface for writing, so there are some variations of depth due to the writing process which includes motion-oriented movements of the hand muscles. These variations in depth can be utilized as important features to improve air-writing recognition accuracy. However, considering the index finger movement information in Z-axis may generate an inhomogeneous distribution of smaller depth values degrading the recognition performance. So, converting the actual depth values into a scaled range of varying depth values, homogeneously distributed into certain levels, should give good recognition results.

On-air writing makes the writing process natural, unconstrained, and at the same time challenging. When someone writes an alphabet, she/he writes it as a sequence of strokes to represent the English alphabets. The best algorithm for air-writing should be able to segment the strokes accurately from the air gestures. However, in air-writing, many extra movements of the user match with perfect strokes [1] and hence become part of the writing.

In this paper, we propose a system to recognize unconstrained air-writing of 26 ECAs. To facilitate unconstrained writing, we did not impose any restrictions on the user, such as 'write slowly' or 'try to write perfectly.' We represented the hand trajectories, that is, the hand movement sequence as a series of data points (x_t, y_t, d_t), where (x_t, y_t) is the position of the hand and d_t is the depth value at time sequence t. The depth values are quantized at certain levels. Those data points are converted into the time-series representation of a particular alphabet suitable for extracting features. We have determined 12 time-series features and represented those as point vectors (x and y dimensions), the depth value of the corresponding point, quantized depth value, pointwise distances, theta value, velocity, log-normal probability density functions (mean and standard deviation), freeman chain codes (4, 8, and 16). We have generated those 12 features for each alphabet from 22 users. Out of them, data from 15 users were used to generate DTW distance features, and we found data from 7 users are almost perfect as we expected to consider them as templates for the DTW algorithm. Hence, one best data for each alphabet was taken manually as a template from 7 users apart from those 15 users. After normalizing those distance values, we fed them into a multiclass SVM classifier. The main research contribution of this study are as follows:

1. Generation of a unique depth-based air-writing dataset consisting of 26 ECAs in an unrestricted environment.
2. Introducing DTW-based distance features for air-writing. The index finger movement trajectory was captured using depth information while writing an ECA letter and represented as time-series data.
3. Utilizing the depth information as significant features (depth value provided by Kinect as one feature and the quantized depth value as another feature) to capture the motion-oriented movement of the hand while performing natural writing in the air.

This paper is the extended work of our previous research in [6], where we used only DTW-based classification considering data from one user with 5 variations. In that paper, we did not consider DTW distances as features for a multiclass classifier. However, in this paper, we took ECA gestures from 15 users, created a larger data set, and determined the DTW distance features for SVM training and testing. Moreover, we have utilized the depth information as significant features which contributed to the improvement of recognition accuracy.

2 Related work

Human gesture is an important input modality for communication with computers in designing gesture-based interfaces. A typical hand gesture recognition system uses a camera (typical stereo camera) to read the hand movement data, performs the hand tracking, and then recognizes a meaningful gesture to control any devices or applications.

Air writing means gesture-based writing on the air through movement of hand fingers by which a computer system can recognize language-specific characters and other symbols in natural handwriting [1]. In the process of air-writing, each movement of the hand becomes a stroke. So, alongside the actual writing, many noises are introduced into the writing. The authors in [1] defined English characters as a sequence of strokes. The capital alphabet 'A,' for instance, is composed of three strokes mainly '/,' '\' and '-.' If the discrete strokes can be pulled out from the seemingly continuous movement of the hand, it is possible to infer the characters. The basic set of strokes for constructing the alphabet is shown in Fig. 1.

In [2], the researchers showed how a wearable device can recognize hand gestures for air writing. The air-writing glove fits at the back of the hand. It has motion sensors, accelerometers, and angular rate sensors equipped with a smartphone. The signals are recorded and transmitted via

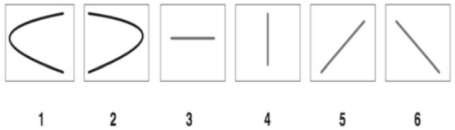


Fig. 1 Basic strokes for English characters

Bluetooth. A wearable hand motion tracking system captures movement signals using an accelerometer and gyroscope. However, converting the acceleration signal into important features to recognize strokes can be erroneous due to the drift of inertial sensors. Moreover, wearing a special device makes the air writing system cumbersome and not natural. Sensors attached to a glove record hand movements; a computer system captures relevant signals and translates them into text, which can then create an email, text message, or any other type of mobile app [10].

Yin et al. [20] use an online approach, attentive contexts-vector (AC-Vec), and an offline approach, attentive contexts-convolutional neural network (AC-CNN), for character recognition. Kim et al. [9] showed a way to recognize different people's handwriting on continuous images based on the similarity of the different shapes of characters or digits based on the strokes and the ligature model. They did not use the concept of bare handwriting without using any special input pen. They tried to generate virtual 3D characters from 2D shapes using the ligature model and then used the Bayesian model to recognize real on-air writing. In our approach, we are using an unconstrained environment to write English alphabets, creating a training model using real on-air writing gestures. We are using the character shape and movement information as features in the form of time-series curves. On-air writing alphabets can be considered as signals produced at a particular time duration. So the alphabets are special curves with time variations. For example, a person can take 3 s to write the character 'A' and another person may take 5 s to write the same. Dynamic Time Warping (DTW) is a popular technique for matching variable-length signals, and the DTW algorithm is able to compare two curves in a way that makes sense and helps in matching the same patterns of the curves [12].

Researchers in [15] tried to recognize air-written Persian digits representing numbers from 0 to 9. They have addressed the research issues related to ligature stroke. The gradient variations on the trajectories are used as features for the recognition task. Though they said these features are scale, rotation, and translation invariant, but there is a scope of further investigations considering the scale, rotation, translation-invariant features provided by the algorithms like scale-invariant feature transform (SIFT), Speeded up robust features (SURF), Oriented FAST and rotated BRIEF (ORB), Gradient location and orientation histogram (GLOH), etc [11]. They have implemented an analytical classifier and compared the results with other state-of-the-art classifiers.

In [11], the researchers worked on symbolic hand gesture recognition and tried to recognize hand postures of 0–9 numeric symbols using depth information. They tried to extract informative SIFT features from contrast varying depth image, and the contrast variation was performed through depth-map quantization. They hypothesized that the depth-map quantization process can give better recognition accuracy which was not previously explored in static hand gesture recognition. They applied the idea in the benchmark static hand gesture dataset and got better recognition accuracy compared finger-earth mover's distance (FEMD)-based [14] method. However, we found a scope to utilize the depth-map information effectively in air-writing recognition. Section 3.3 contains the description of our selected set of features for on-air writing recognition.

3 Proposed approach

Any recognition system must have data collection, i.e., image Acquisition, preprocessing step which may include segmentation, feature extraction, and then classification. Sometimes a post-processing step may be required before classification for feature dimension reduction, feature selection for further analysis. Figure 2 shows an overview of our proposed system.

3.1 Image acquisition

We placed a depth camera (Microsoft Kinect) in front of the individual user and asked to write an upper-case English letter considering an imaginary writing board. All the users tried to apply their self-writing style so the font-size and speed of writing highly varied. This caused the dataset very much challenging in terms of feature generation for training and testing the classification model. We asked every user to write from 'A' to 'Z' in a sequence one after another. Then we have isolated every alphabet, with the help of depth and RGB values found on the gesturing image signal. Usually, a user pause writing two consecutive letters that gives the user feel comfortable. We were able to accumulate ECA data from 22 users. However, we have manually observed, contemplated, and analyzed the individual ECA written by each user and found the best ECA data from 7 users to be considered as a template that we have mentioned earlier in Sect. 1. The rest of the ECA from 15 users were used for distance feature generation.

3.2 Segmentation and pre-processing step

We assume the hand is the front-most object while the user is writing on the air. The pre-processing steps were as follows: at first, the hand from the background was separated by using depth information which is the part of hand segmentation process as shown in Fig. 3c.

From the segmented hand image, we have extracted the x, y coordinate of a point in that image by calculating the middle pixel location between the starting and the end position that contains nonzero pixel value. We have followed this process

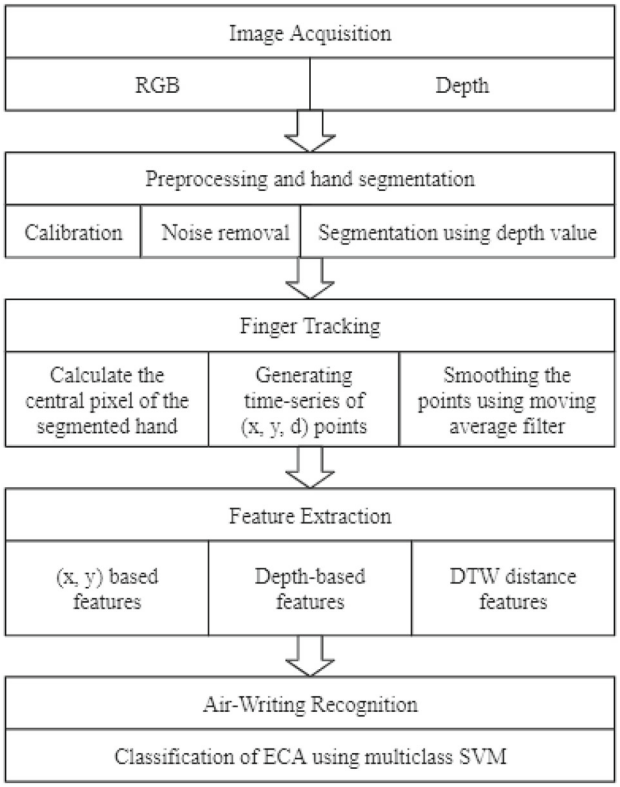


Fig. 2 Proposed approach

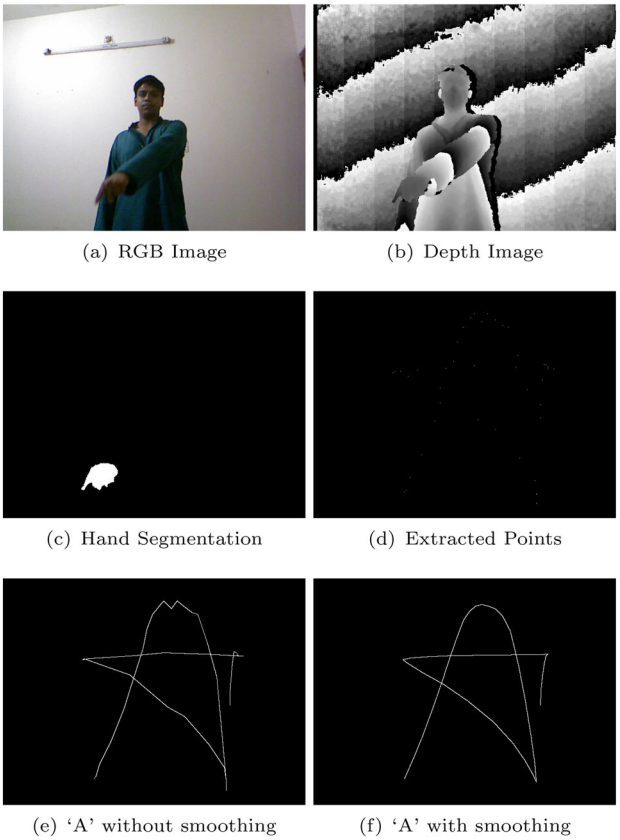


Fig. 3 Air-writing process to generate the letter 'A'

for each image frame and got the consecutive points of the hand movements as shown in Fig. 3d.

We have considered these middle points while writing a letter and generated an image consisting of that letter as shown in Fig. 3e. These points contain hand-movement trajectory location in the process of air-writing and the number of consecutive points represent the time-series data. So for writing each letter, we have traced out the written points (x, y) and their corresponding depth values (d), represented as points (x_t, y_t, d_t) at time t. Tracking the hand motion from image to image gave us a series of points (x_t, y_t, d_t). Those sets of points are the time-series information of a particular alphabet.

Air-writing process may lead to uncontrollable jerky movement [18] of the hand fingers. We found the written letters are not in a legitimate shape. Hence, the generated raw image is smoothed using a moving average filter [5]. The written letter after smoothing is shown in Fig. 3f. The overall process of writing the ECA letter 'A' is shown in Fig. 3. The corresponding time series curve is given in Fig. 4. In Fig. 4, the X-Axis represents the consecutive points in total 3 s taken to write the ECA, 'A' by a particular user and the Y-axis represents the corresponding pixel values in the gesturing image.

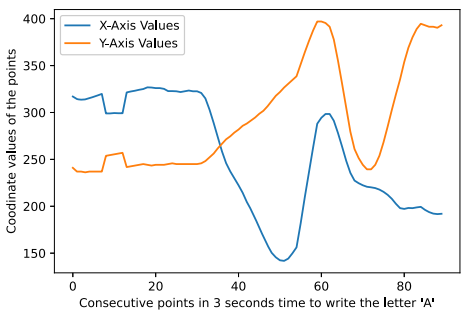


Fig. 4 Time-series representation of 'A'

3.3 Feature extraction and classification

After converting the air written alphabet to a time-series of x, y, and d, the task is to classify them. As finding a stroke feature proved to be very difficult, we propose to classifying them based on time-series data. So, we investigated the use of DTW as the classifier. Our earlier work in [6] was about matching 2D trajectory (x, y) of an alphabet with templates and come up with a decision based on DTW distance using Eq. 1.

$$ClassifiedClassLabel(trajectory(x, y)) = argmin(dist(trajectory(x_{template}, y_{template}), trajectory(x, y))) \quad (1)$$

Here, we get the minimum DTW distance between the template trajectories and (x, y) whose class is being identified.

The decision taken from the DTW distances was not that accurate with a small number of users [6]. When we increased the number of users from 5 to 15, the accuracy reduced to half. Then we looked for other features besides point vectors such as pointwise distance, theta value of points, velocity, log-normal probability density, and freeman chain code that are used regularly in online handwriting recognition. We have also included quantized depth information where the depth value was converted into a range of 155–255 and 10 levels. The quantization process that we have followed is the same as shown in [11]. Each of the depth value for an ECA was quantized using 2

$$Q(Z) = \mathrm{DL}_{\min} + \left(\left\lfloor \left(\frac{D(Z) - D_{\min}}{D_{\mathrm{th}} - D_{\min}} \times \eta \right) + 0.5 \right\rfloor \times \left\lfloor \frac{\mathrm{DL}_{\max} - \mathrm{DL}_{\min}}{\eta} \right\rfloor \right) \quad (2)$$

Here, $D(Z)$ is the distance of point at the (x, y), $Q(Z)$ is the quantized depth value of the corresponding depth-map. η is the quantization levels between $\mathrm{DL}_{\min}$ and $\mathrm{DL}_{\max}$. The movement of the fingers that we have considered is the distance values between $D_{\min}$ and the depth threshold D_{th} within which the hand finger movements are found.

In total 12 features were represented as time-series information. An example time-series representation is shown in Fig. 5 and in Fig. 6. The list of features is given in Table 1. The DTW distances of the 12 time-series features were compared with the alphabet templates and directly used for classification. Still, the result was not significant to report. Phase shift in signals reduces the accuracy of recognition as the DTW algorithm does always care about the phase differences [8]. However, the geometric shape information may be required to preserve as important features to learn [7]. In such a situation, the state-of-the-art analysis suggested us to use all-pair comparison and use the DTW distance features for learning with another classifier. We choose SVM as a multi-class classifier for this purpose.

In Fig. 5, we show the kinect-image pixel value including depth image for particular characters. Similarly Fig. 6 shows a simple derivative feature from X and Y axis. Here we selected pointwise distance.

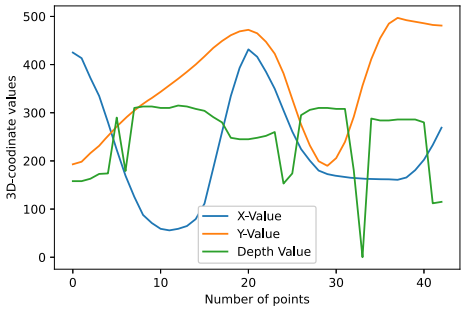


Fig. 5 Time-series of point vector and the depth value for the letter, 'A'

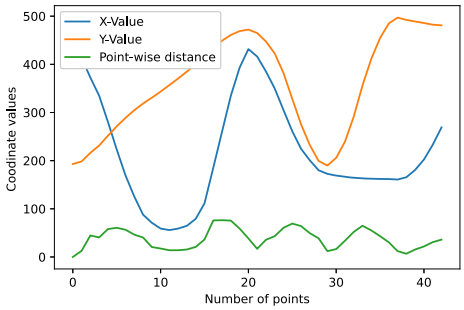


Fig. 6 Time-series of point vectors and the pointwise distance values for the letter 'A'

3.3.1 DTW distances as derived features

After separating template and user data from the entire dataset we have converted every image to a time-series curve or signal which is shown in Fig. 4. In the x-axis, the consecutive points that we have extracted as the point vector within a particular time in seconds are arranged to represent time-series data. At 30fps, different users have taken different amounts of time to write the same letter. In Fig. 4, we have shown the time-series data generated from 90 frames which took 3 s to write the English Capital Alphabet (ECA), 'A.' We have represented every signal to a feature vector for each ECA. The list of 12 features is given in Table 1.

DTW gives us minimum distances between two time-series curves. When a user writes an alphabet in an imaginary blackboard, the user does not necessarily round-up with the same length of input for the alphabets. It also varies in case of writing the same alphabet by different users. DTW algorithm is the right one to apply in this scenario so that we can find the minimum distance between the two alphabets.

In our proposed approach, we compare an alphabet represented as a point vector, to all alphabet's point vectors using the DTW algorithm. That means, 12 time-series features of an alphabet were compared with corresponding features of the template which gives 12 distance values. Comparing an alphabet with all 26 templates generate $12 \times 26 = 312$ distance features. We have used basic DTW equations to calculate the distances. The class label for these 312 dis-

Table 1 Features used for on-air handwriting recognition

Features	Description
Feature 1 and Feature 2 (F_1, F_2): Point vector of alphabets	The point vector generates 2 time series features: one for x-dimension and one for y-dimension
Feature 3 (F_3): Depth value of the point	Depth value was extracted from the hand trajectory and smoothed. As there as fewer movements in the depth, other derived features such as velocity was not calculated from the depth information
Feature 4 (F_4): Quantized depth value	The quantized depth value within 155–255 using Eq. 2
Feature 5 (F_5): Pointwise distance of point vector	This is the euclidean distance of consecutive two trajectory points (x, y)
Feature 6 (F_6): Theta value of point	This feature helps to measure pixel-wise angular distances in polar coordinate
Feature 7 (F_7): Velocity of point	This feature helps to generate data point from pointwise distance which shows the speed within that distance either forward or backward
Feature 8 and Feature 9 (F_8, F_9): Lognormal probability density function calculation mean and standard deviation of data point	This function is calculated based on the average and standard deviation. The point vector that we are taking is based on time sequences which are always positive. So, the log-normal distribution function of the two dimensions will always give nonnegative values. Moreover, alphabet writing does not follow normal distribution, so log-normal distribution can be a good feature for time-series classification
Feature 10, Feature 11, Feature 12 (F_{10}, F_{11}, F_{12}): Freeman chain code of 4, 8, 16	Freeman chain code is a shape-based matching technique found to be successful in recognizing digits or characters [3].Chain code represents the sequence of direction changes between adjacent points of a curve. In [17], freeman directional code was generated for dynamic hand gesture for recognition

tance features is given as the alphabet under consideration. We define the set of users as S, where, $|S| = 15$. For our 15 users, we have calculated 12 DTW distances (between 12 source features and 12 template features for each alphabet) to produce a 312-dimensional feature vector and normalized them between 0 and 1. So, each user writing 26 letters produces $15 \times 26 = 390$ instances and 312-dimensional feature per instance. Let Σ be the time-series representation of ECA characters 'A' to 'Z' generated by each user $s \in S$, i.e.,

$$\Sigma = \{U^A,\ U^B,\ U^C, \ldots, U^Z\} \tag{3}$$

and T be the time-series representation of the template of each alphabet, i.e.,

$$T = \{T^A,\ T^B,\ \ T^C \ldots,\ T^Z\} \tag{4}$$

Each user generated ECA character and the template ECA character contains 12 features, i.e.,

$$F = \{F_1,\ F_2,\ F_3 \ldots,\ F_{12}\} \tag{5}$$

where features in the user generated ECA and template ECA possibly have different lengths. Each feature consists of normalized values from 0 to 1.

$$F_i = \{x \in R, 0 \leq x \leq 1, \forall i \in [1, 12]\} \tag{6}$$

We determine the pair-wise minimum DTW distance between U^A and T^A, between U^A and T^B, and so on up to between U^A and T^Z for all the 12 features. These are the distance features generated by taking the distances between U^A and all the template elements of T. This makes the first instance of the first user writing ECA character 'A' which we denote as follows:

$$S_1^A = [\text{DTW}(U^A_{F_{1\ldots12}}, T^A_{F_{1\ldots12}}), \text{DTW}(U^A_{F_{1\ldots12}}, T^B_{F_{1\ldots12}}), \ldots \text{DTW}(U^A_{F_{1\ldots12}}, T^Z_{F_{1\ldots12}})] \tag{7}$$

Let $\text{DU}^{A,A}_{F_{1\ldots12}}$ are the DTW distance features between U^A and T^A, $\text{DU}^{A,B}_{F_{1\ldots12}}$ be the DTW distance features between U^A and T^B, and so on. We continue to generate these 12 features for each character up to ECA character 'Z' and the features for the last DTW distance between U^A and T^Z are $\text{DU}^{A,Z}_{F_{1\ldots12}}$. Thus we get $12 \times 26 = 312$ features for the first user writing ECA character 'A.'

So, for the first user generating 26 ECA characters we get 26 samples and the first training sample is S_1^A, the second training sample is S_1^B and so on. For all the 15 users, $\tau^{A\ldots Z}$ will produce a training set of size 390×312 with 26 class labels. We can represent this training set using the matrix as per Eq. (9).

$$S_{i\in S}^{A\ldots Z} = \begin{Bmatrix} \mathrm{DU}_{F_{1\ldots12}}^{A,A} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{A,M} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{A,Z} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \mathrm{DU}_{F_{1\ldots12}}^{M,A} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{M,M} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{M,Z} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \mathrm{DU}_{F_{1\ldots12}}^{Z,A} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{Z,M} & \ldots & \mathrm{DU}_{F_{1\ldots12}}^{Z,Z} \end{Bmatrix} \tag{8}$$

$$\tau^{A\ldots Z} = \begin{Bmatrix} S_1^{A\ldots Z} \\ \ldots \\ S_7^{A\ldots Z} \\ \ldots \\ S_{15}^{A\ldots Z} \end{Bmatrix} \tag{9}$$

4 Experimental result and evaluation

We have evaluated the proposed system in our own generated air-writing dataset consisting of 26 ECAs gestures performed by 22 users (all are male). There were no pre-instruction or guidelines on font size, speed of writing which makes the dataset more challenging. The number of samples per ECA varies among different users as we can see in Fig. 7.

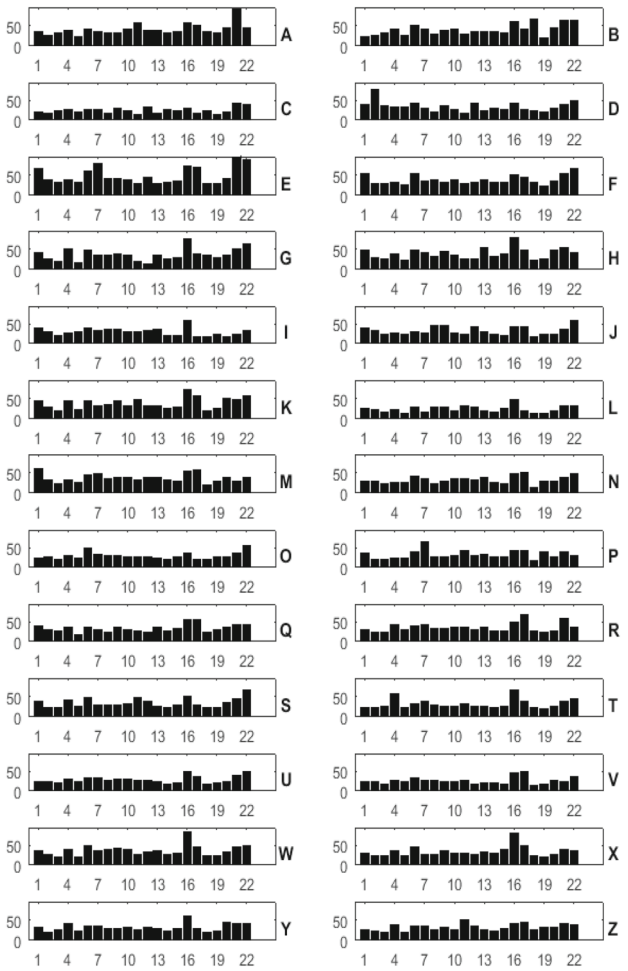


Fig. 7 Sample distribution of 22 users for each ECA (A to Z) where *x*-axis represents user number and *y*-axis represents the number of samples per user

Table 2 Information on the speed of ECA air-writing by 22 users in seconds

ECA	Avg	Std	Max	Min	ECA	Avg	Std	Max	Min
A	1.41	0.48	3.00	0.73	N	1.07	0.28	1.67	0.43
B	1.42	0.49	2.23	0.67	O	1.01	0.29	1.83	0.63
C	0.92	0.33	1.70	0.50	P	1.10	0.33	2.17	0.57
D	1.19	0.43	2.63	0.63	Q	1.22	0.41	2.13	0.60
E	1.69	0.69	3.40	0.97	R	1.22	0.37	2.27	0.77
F	1.38	0.45	2.43	0.73	S	1.19	0.39	2.20	0.73
G	1.33	0.56	2.53	0.50	T	1.06	0.36	2.17	0.63
H	1.40	0.50	2.60	0.77	U	1.02	0.34	1.97	0.57
I	1.01	0.30	1.97	0.57	U	1.02	0.34	1.97	0.57
J	1.14	0.36	1.93	0.63	W	1.27	0.44	2.83	0.70
K	1.34	0.45	2.43	0.67	X	1.12	0.40	2.70	0.67
L	0.86	0.28	1.53	0.43	Y	1.16	0.48	3.00	0.67
M	1.12	0.32	2.00	0.70	Z	1.13	0.32	1.80	0.63

The maximum number of samples (102) required to write an ECA is 'E' while the minimum number of samples required to write the ECA characters are 'I' and 'L' by most of the users. A total of 19,350 samples were collected to build the ECA dataset. Moreover, if we analyze the writing speed as given in Table 2, there were also variations in the duration of writing the same ECA by different users. To write 'E,' the average number of users required the maximum amount of time 1.69 s whereas the minimum 0.86 s and 1.01 s time required to write 'L' and 'I,' respectively.

From all the samples, features were extracted as described in Sect. 3.3. To prepare the training set for the classification, we need unified dimensional feature vectors. However, we have considered DTW distances as features for training and testing considering the following reasons:

1. Air-writing is a temporal activity that can be represented as time-series data but due to variation of movement time, while writing, the length of two ECA varies.
2. Variable-length time-series values can be represented as features if they are in fixed-size; DTW distance features generated using Eq.(9) gave us a unified dimensional feature vector for training and testing.
3. All-pair comparison of the features among ECAs helps the classifier to learn the information related to phase differences.

The dataset contains 312 DTW distance features with 390 instances. However, 25% and 50% resampling applied in the dataset gave us 468 and 572 instances respectfully. To analyze the impact of depth information in recognition results,

Table 3 12 datasets for air-written ECA recognition

Datasets	Dimensions	Dataset Description
Dataset 1	390 × 260	Without depth information
Dataset 2	390 × 312	With depth information
Dataset 3	468 × 260	Without depth, 25% resampled
Dataset 4	468 × 312	With depth, 25% resampled
Dataset 5	572 × 260	Without depth, 50% resampled
Dataset 6	572 × 312	With depth, 50% resampled
Dataset 7	390 × 120	Without depth, attribute selected using correlation
Dataset 8	390 × 155	With depth, attribute selected using correlation
Dataset 9	390 × 135	Without depth, attribute selected using Information Gain
Dataset 10	390 × 171	With depth, attribute selected using Information Gain
Dataset 11	390 × 139	Without depth, attribute selected using Gain Ratio
Dataset 12	390 × 171	With depth, attribute selected using Gain Ratio

we have prepared two sets of datasets, one is without depth information (390×260) and the other one is with depth information (390×312) including their two sets of re-sampled versions (468×260 and 572×260; 468×312 and 572×312). Moreover, we have prepared three sets of dataset containing features related to correlation analysis. Thus we have prepared 12 datasets to conduct the evaluation. The description of the datasets is given in Table 3. So, in general we can divide our dataset in to two groups: Dataset with depth information (in Table 3, Dataset 1, 3, 5, 7, 9, and 11) and datasets without depth information (in Table 3, Dataset 2, 4, 6, 8, 10, and 12). We have used 12 datasets to understand the significance of depth information from different perspectives, like, taking all the features, taking only the depth features, taking the re-sampled features, taking features after correlation analysis.

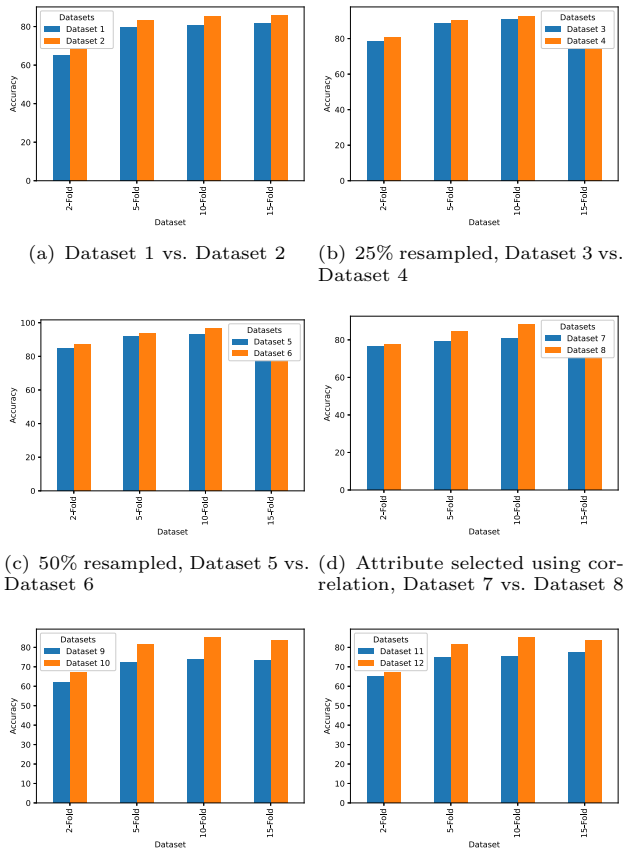
The cross-validation process was performed by splitting the dataset into k-folds to train the model on all the samples except the $(k-1)$ folds and evaluated the model on the fold that was not used for training. This process was repeated k-times and recorded the average accuracy. 2-Fold, 5-Fold, 10-Fold, and 15-Fold cross-validations were performed in the datasets, and the recognition accuracy is recorded. All the experimental evaluations were conducted on the Intel Core I7 2.60 GHz CPU of 16 GB RAM. We collected hand gesture images of each ECA at a resolution of 640 × 480 pixels at 30 fps with the help of Kinect in a C# and Microsoft dot net based system. All the pre-processing steps, finger tracking, feature extraction, and dataset preparation for machine learning were implemented using Matlab software. We perform machine learning analysis using Weka [4] software.

To measure the classification accuracy we have determined true positive rate (TPR), false positive rate (FPR), precision, recall, F-measure, receiver operating characteristics (ROC) area value, and precision-recall curve (PRC) area value. The classifier performed well as we can see

Table 4 Confusion matrix of dataset 8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	10	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0
E	0	1	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
G	0	0	0	0	0	0	12	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	2	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	1	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	1	1	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	13	2	0	0	0	0	0	0	0	0	0	0	0	0
N	0	1	0	1	0	0	0	0	0	0	0	0	1	12	0	0	0	0	0	0	0	0	0	0	0	0
O	1	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	1	0	0	0
S	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0	0	0	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	13	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	2	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	3	0	0
X	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	13	0
Z	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	11

the average ROC area value for 10-Fold cross-validation is within 0.9–1. Moreover, the average PRC value for all the datasets is between 0.8 and 0.9. The classification result for dataset 8 has been summarized in the confusion matrix as given in Table 4. This dataset contains the minimum number of features with depth information compared to other dataset containing depth features. Moreover, after the dataset 6 (572 × 312, 50% resampled, accuracy 96.85%), this is the dataset with depth features for which we got the maximum accuracy (88.2%) in 10-fold cross-validation. The average accuracies that we have got for TPR, FPR, precision, recall, F-measure, ROC, and PRC are 88.2%, 0.5%, 89.6%, 88.2%, 88.4%, 99.5%, and 93.9%, respectively. We got the highest F-Measure score for the letters 'C,' 'H,' 'I,' and 'Q' which is 100% and lowest 66.11% for the letter 'N.' The result is generated considering 155 normalized DTW distance features.



(a) Dataset 1 vs. Dataset 2

(b) 25% resampled, Dataset 3 vs. Dataset 4

(c) 50% resampled, Dataset 5 vs. Dataset 6

(d) Attribute selected using correlation, Dataset 7 vs. Dataset 8

(e) Attribute selected using Information Gain, Dataset 9 vs. Dataset 10

(f) Attribute selected using Gain Ratio, Dataset 11 vs. Dataset 12

Fig. 8 Comparison of cross-validation results of the 12 datasets grouped by without depth features and with depth features as described in Table 3 where x-axis represents cross-validation folds and y-axis is represents accuracy

We can see the cross-validation comparison results of the prepared 12 datasets divided into two groups: Dataset with depth information (in Table 3, Dataset 1, 3, 5, 7, 9, and 11) and datasets without depth information (in Table 3, Dataset 2, 4, 6, 8, 10, and 12). We have used these 12 datasets to understand the significance of depth information from different perspectives, like, taking all the features, taking only the depth features, taking the re-sampled features, taking features after correlation analysis. After performing the k-fold cross-validation of each of the datasets, we were able to achieve higher accuracy for depth-feature-based datasets. The comparison results are shown in Fig. 8. The highest accuracy we were able to achieve is 96.85% for dataset 6 for which the dataset was generated by taking random subsamples from the original dataset (dataset 2) with replacement. The datasets considering depth information always gave better accuracies compared to datasets without depth information. Taking all the features gave us 9.16%, 5.16%, 5.40%, and 5.35% accuracy improvements in 2-Fold, 5-Fold, 10-Fold, and 15-Fold cross-validations, respectively, over the datasets that do not contain depth features.

We tried to understand the impact of different features to justify our recognition accuracies. All the features except F_3 and F_4 are derived features from F_1 and F_2. The features F_1 and F_2 only gave us the recognition accuracies of 25.1282%, 27.6923%, 29.3208% and 30.7992% for 2-Fold, 5-Fold, 10-Fold and 15-Fold cross-validations, respectively. However, if we add one-by-one feature the accuracies improved significantly. For example, if we add features F_5, F_6, F_7, F_8, F_9, and F_{10} with feature F_1, F_2 which are without depth features F_3, F_4, we got 157%, 181%, 169%, 160% improvement and with depth features we got 173%, 197%, 182%, 171% (for 2-Fold, 5-Fold, 10-Fold, 15-Fold cross-validations, respectively) improvements. We justified the use of taking 3 freeman chain code features F_{10}, F_{11}, and F_{12}, we wanted to take only one feature out of these three features. However, taking three features gave us overall accuracy improvement around 2.62% compared to taking any one feature.

We applied feature selection techniques to cope with overfitting problems and ranked the features based on information gain, gain ratio, Pearson's correlation. We remove features that do not contain significant information. We tried to understand the relationship between different features and their corresponding class labels and tried to analyze which features and how many features contribute more to recognition accuracy. We found that features with depth information contribute more to recognition accuracy for all the feature selection techniques we applied. In the case of information gain and the gain ratio of the attribute with respect to class, we got 48 features with a ranked value greater than 0, resulted in 73.33% accuracy. Removing 100 worst-ranked features for both of the techniques gave us 80.77% accuracy. However, we got the highest accuracy by removing 141 features starting from the last, which means, using the top 171 features we found the highest accuracy 85.13% with depth features. We removed the features without depth information within these 171 features and got the highest 73.85% accuracy out of 135 features. We tried to select the features based on Pearson's correlation coefficient values, with cut-off value 0.07 gave us 155 features, 88.21% accuracy with depth features, and 82.05% accuracy using 120 features without depth features. After this empirical analysis, we found that in the case of information gain and gain ratio methods, the difference between with-depth features and without-depth features is 36 and 32, respectively, whereas in case of Pearson correlation the difference is 35. However, information gain or gain ratio-based methods gave us 171 features including depth features and Person correlation method gave us 155 features. So, we retain a minimum number of features with depth values using Pearson correlation technique in case of dataset 8 which gave us the highest recognition accuracy. We found that features

with depth information always gives better results compared to features without depth information. The highest difference we got for dataset 10 is 15% and the lowest difference we got for dataset 4 is 2% and in an average, for all the datasets we got 7% difference in 10-Fold cross-validation.

5 Conclusion

In this paper, we tried to recognize on-air hand-written characters of English Capital Alphabets (ECAs) through a Kinect depth camera. It is a vision-based spatio-temporal activity in which the hand trajectory vectors were generated and utilized for each of the gesturing images. We have created a unique dataset in a complex natural environment with the help of 15 users. Each of the ECAs is presented as time-series values containing 12 discriminating features. Then, all pair DTW distances were calculated and a total of 312 distance features represented each of the alphabets. We got 390 instances containing the 312 feature dimensions for SVM training and testing. However, we also analyzed the recognition accuracy by removing features from the ranked feature list based on information gain, gain ratio, and correlation analysis. With these, we have generated 12-datasets with a different number of features based on feature analysis. We have performed 2-Fold, 5-Fold, 10-Fold, and 15-fold cross-validation and found high recognition accuracy of 96.84% by resampling instances and also 88.21% using 155 ranked feature list based on feature correlation analysis for our selected number of features. These results we have achieved considering depth information as important features compared to non-depth features. In the future, we will continue our work to recognize small-letter English alphabets as well as Bangla alphabets including word recognition.

Acknowledgements The authors would like to thank to all users who have participated voluntarily in the data collection process.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Agrawal, S., Constandache, I., Gaonkar, S., Roy Choudhury, R., Caves, K., DeRuyter, F.: Using mobile phones to write in air. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys '11, New York, NY, USA, pp. 15–28. ACM (2011)
2. Amma, C., Georgi, M., Schultz, T.: Airwriting: a wearable handwriting recognition system. Pers. Ubiquitous Comput. **18**(1), 191–203 (2014)
3. Freeman, D., Barrentine, D.B.: Method and system for operating a multi-function portable electronic device using voice-activation, March 10 . US Patent 8,977,255 (2015)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. **11**(1), 10–18 (2009)
5. Hunter, J.S.: The exponentially weighted moving average. J. Qual. Technol. **18**(4), 203–210 (1986)
6. Islam, R., Mahmud, H., Hasan, M.K., Rubaiyeat, H.A.: Alphabet recognition in air writing using depth information. In: The Ninth International Conference on Advances in Computer–Human Interactions, pp. 299–301 (2016)
7. Jalalian, Arash, Chalup, Stephan K.: GDTW-P-SVMS: variable-length time series analysis using support vector machines. Neurocomputing **99**, 270–282 (2013)
8. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recognit. **44**(9), 2231–2240 (2011)
9. Kim, D.H., Choi, H.I., Kim, J.H.: 3d space handwriting recognition with ligature model. In: Proceedings of the Third International Conference on Ubiquitous Computing Systems, UCS'06, pp. 41–56. Springer, Berlin (2006)
10. Knowledge Euronews. Future of texting: writing in the air! Apr 8, 2013
11. Mahmud, H., Hasan, M., Kabir, M., Mottalib, M.A.: Recognition of symbolic gestures using depth information. Adv. Hum. Comput. Interact. (2018). https://doi.org/10.1155/2018/1069823
12. Mullin, R.: Time warps, string edits, and macromolecules. In: Kruskall, J., Liberman, M. (eds.) The theory and practice of sequence comparison. Advanced book program, Addison-Wesley, Reading, MA, Don mills, ON, 300 pp. us $31.95. ISBN: 0-201-07809-0 (1983). Can. J. Stat. 13(2):167–168 (1985)
13. Priya, A., Mishra, S., Raj, S., Mandal, S., Datta, S.: Online and offline character recognition: a survey. In: International Conference on Communication and Signal Processing (ICCSP). IEEE (2016)
14. Ren, Z., Yuan, J., Meng, J., Zhang, Z.: Robust part-based hand gesture recognition using kinect sensor. IEEE Trans. Multimed. **15**(5), 1110–1120 (2013)
15. Mohammadi, S., Maleki, R.: Air-writing recognition system for Persian numbers with a novel classifier. Vis. Comput. **36**, 1001–1015 (2019)
16. Siena, F.L., Byrom, B., Watts, P., Breedon, P.: Utilising the intel realsense camera for measuring health outcomes in clinical research. J. Med. Syst. **42**(3), 53 (2018)
17. Sreekanth, N.S., Narayanan, N.K.: Dynamic gesture recognition: a machine vision based approach. In: Proceedings of the International Conference on Signal, Networks, Computing, and Systems, pp. 105–115. Springer (2017)
18. Thomas, M.: A role for "air writing" in second-language learners' acquisition of Japanese in the age of the word processor. J. Jpn. Linguist. **30**(1), 86–106 (2014)
19. Wexelblat, A.: An approach to natural gesture in virtual environments. ACM Trans. Comput. Hum. Interact. **2**(3), 179–200 (1995)
20. Yin, Y., Xie, L., Gu, T., Lu, Y., Lu, S.: AirContour: building contour-based model for in-air writing gesture recognition. ACM Trans. Sens. Netw. **15**(4), 44:1–44:25 (2019)

Hasan Mahmud is an Assistant Professor for human-computer interaction in CSE department of IUT, a subsidiary organ of OIC. He has been involved in HCI research since 2009. His specialization is in the area of gesture-based interaction through machine learning approaches, affective computing, and assistive technology for the physically impaired.

Robiul Islam has recently joined as a Ph.D. student at Innopolis University, Russia, in neuroscience and cognitive technology laboratory. Previously, he completed his masters degrees in 'System and Software Engineering' at National Research University Higher School of Economics (HSE), Russia, and 'Computer Science and Engineering' from the Islamic University of Technology (IUT), Bangladesh. He was a research assistant at Laboratory of Computational Pragmatics Models and Methods, HSE. His research interest is in machine/deep learning, human brain-computer interaction, computer vision, and visualization.

Md. Kamrul Hasan has received his Ph.D. from Kyung Hee University, South Korea. Currently, he is working as a Professor of CSE, IUT, OIC. He has expertise in intelligent systems and AI, software engineering, data mining applications, and social networking. He is the founding director of SSL research lab, IUT.