

Selection of features and hidden Markov model parameters for English word recognition from Leap Motion air-writing trajectories

Deval Verma¹ | Himanshu Agarwal²  | Amrish Kumar Aggarwal²

¹School of Computer Science & Engineering, Bennett University (Times Group), Greater Noida, Uttar Pradesh, India

²Department of Mathematics, Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India

Correspondence

Himanshu Agarwal, Department of Mathematics, Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.

Email: himanshu.agarwal@jiit.ac.in

Abstract

Air-writing recognition is relevant in areas such as natural human–computer interaction, augmented reality, and virtual reality. A trajectory is the most natural way to represent air writing. We analyze the recognition accuracy of words written in air considering five features, namely, writing direction, curvature, trajectory, orthocenter, and ellipsoid, as well as different parameters of a hidden Markov model classifier. Experiments were performed on two representative datasets, whose sample trajectories were collected using a Leap Motion Controller from a fingertip performing air writing. Dataset D_1 contains 840 English words from 21 classes, and dataset D_2 contains 1600 English words from 40 classes. A genetic algorithm was combined with a hidden Markov model classifier to obtain the best subset of features. Combination {trajectory, orthocenter, writing direction, curvature} provided the best feature set, achieving recognition accuracies on datasets D_1 and D_2 of 98.81% and 83.58%, respectively.

KEYWORDS

air writing, classification, feature selection, Leap Motion Controller, word recognition

1 | INTRODUCTION

Three-dimensional (3D) handwriting or air writing consists of writing characters, digits, words, or texts in the air by moving a finger or wearable device. The recognition of air writing is an important and interesting effort to enhance natural human–computer interactions [1], augmented reality, and virtual reality [2]. In addition, this technology may reduce the size of upcoming keyboards and TV controllers [3] because users would not need to type on a keyboard or write on a trackpad/touchscreen [4]. Moreover, air-writing analysis can be useful in applications related to the recognition, interpretation, and

identification of conventional handwriting. Such applications include handwriting learning tools, writer authentication, and signature verification [5–9].

Popular devices for detecting air writing include depth sensors (e.g., Microsoft Kinect [10], Leap Motion controller—LMC [11], and Intel RealSense 3D depth camera), web cameras, WiFi inertial sensors (e.g., accelerometer, gyroscope, magnetometer, wireless inertial measurement unit incorporating accelerometer, gyroscope, and magnetometer [12]), and radars [13–15]. A comprehensive review of the applications and various features (e.g., accuracy, precision, field of vision, robustness, manufacturer, technology, captor range, cost, and

availability) of depth sensors is available in [16]. A device based on inertial sensors should be worn or held in hand while the users move their fingers over the viewing field of the depth sensors or cameras. Recently developed smart bands [17] are user-friendly devices for air-writing.

Hand gestures and trajectory identification are the two main principles of air-writing recognition. For instance, camera-based sensing allows gesture identification from a sequence of images followed by feature extraction [18]. Using a Kinect sensor, a sequence of depth images can be obtained to track a fingertip motion [19–23]. An LMC tracks the 3D locations of fingertips and joints with 0.01 mm accuracy using an application programming interface [24–28]. Misra and others [29] placed a red marker on an index fingertip for air-writing recognition. Among inertial sensors, accelerometers, gyroscopes, and magnetometers provide accelerations, angular velocities, and magnetic signals, respectively, allowing the estimation of hand trajectories [3,30–32]. The Kinect sensor is suitable for tracking whole-body motion, whereas the LMC accurately estimates fingertip motion. Inertial sensors can be more accurate than depth sensors. Nevertheless, the accurate fingertip motion detection and reliable application programming interface of LMC render it the best sensor for capturing air-writing data.

Computer-based handwriting analysis can be divided into three categories: (i) Air writing, (ii) offline two-dimensional (2D) handwriting (data stored as grayscale images [5,6]), and (iii) online 2D handwriting (data stored as 2D spatiotemporal sequence $((x(t), y(t))$ or $(x(t), y(t), f(t))$, where $x(t), y(t)$ are the spatial coordinates, and $f(t)$ is the pen tip force/pressure on the trackpad/touchscreen at time t [6,33]). Existing air-writing data are available in an online format. Thus, their analysis is similar to that of online 2D handwriting but has the following major differences: (i) The start and end positions are unclear; (ii) nonaligned texts are problematic; and (iii) words and text lines have no gaps (i.e., all words belong to a single stroke).

Air-writing recognition can be closely related to offline and online 2D handwriting recognition (with data extended to three dimensions), sound recognition (from spatiotemporal data), and gesture recognition (identification of hand or fingertip motions from an image sequence). Several techniques available for the recognition of 2D handwriting, sounds, and gestures have been extended to air-writing recognition. The four main stages of handwriting analysis include (i) preprocessing [34] (e.g., rotation, size and slant corrections, interpolation, smoothing, resampling, Zernike moment calculation), (ii) segmentation [27,33], (iii) feature extraction [33,34] (e.g., vertical position, curvature, hat, aspect, curliness,

linearity, slope, ascender/descender, context map, normalized coordinate, inflection point, stroke crossing, velocity, directional feature, moment, number of strokes, rendered bitmap, orientation map), and (iv) recognition [33,34]. Recognition methods include the interpretation of time sequences using machine learning models, such as hidden Markov models (HMMs), time-delay neural networks, recurrent neural networks (RNNs), and long short-term memory (LSTM) networks.

Improving 3D air-writing recognition requires fast processing with high accuracy on low-memory devices. Unlike conventional 2D handwriting, 3D air writing shows no gaps between words and text lines [28]. Segmentation of text lines and words can improve accuracy and accelerate recognition. Moreover, 3D air-writing styles vary across writers, who should be trained using well-defined rules (e.g., scripts) for air writing. Two common problems of 3D air writing (but rare in 2D handwriting) are (i) unknown prior knowledge about the start and end positions (even if a common word is written by two different writers or a word is written twice by the same writer) and (ii) nonaligned trajectories given fluctuations in fingertip motion. High-quality trajectories can be recorded by writers skilled in air-writing scripts. Although standards for sign language scripts are available, to the best of our knowledge, no standard for air-writing scripts using the LMC is available. Moreover, the accuracy and computational speed of recognition from a trajectory depend on the selection of features and classification methods. Below, we discuss related work on air-writing recognition using LMC data.

Vikram and others [24] used the LMC to recognize air-written English characters and words. They evaluated the recognition speed and accuracy on spatiotemporal sequences $(x(t), y(t), z(t))$. Dynamic time wrapping was used for character and word recognition.

Xu and others [25] recognized 3755 classes of air-written Chinese characters. They proposed a synthetic method to generate dozens of artificial instances by observing a single instance. The artificial instances supported the training of robust classifiers. Linear discriminant analysis was used for dimensionality reduction, and a compact modified quadratic discriminant function was used for character classification.

Aggarwal and others [26] combined word segmentation and HMM for English sentence recognition. They collected an air-writing dataset (D_1) containing 320 sentences from 10 inexperienced participants writing in the LMC field of view. The words belonged to a set of 21 labeled words, and each word was repeated four times per participant, collecting 840 air-written words. The length of the sentences ranged from two to four words, and the number of characters per word ranged from two

to seven. A partial-differentiation-based technique was used for word segmentation, and the segmented words were recognized by an HMM classifier.

Kumar and others [27] constructed an air-writing dataset (D_2) containing 560 sentences from 10 participants. Each word was drawn 40 times, collecting 1600 words. The sentence length ranged from two to four words, and the number of characters per word ranged from two to eight. A heuristic analysis of the stroke length between two successive words was used for segmentation. HMM and bidirectional LSTM network classifiers were used for recognition.

Kumar and others [28] investigated various features to improve the recognition accuracy on dataset D_1 . Large gaps between the end and start of the lines were used for segmentation. Dynamic and simple features were used by an HMM classifier for recognition. Similarly, Roy and others [35] obtained an average recognition accuracy of 41.7% on a dataset containing 6250 sentences.

The highest accuracy obtained from LMC data for 3D air-written words was 92.7% [28]. Muratha and Shin [21] obtained recognition accuracies of 98% for characters and 95% for digits. Vikram and others [24] focused on the runtime for similarity search of unknown words in two datasets.

The suitability of LMC for air writing motivates the improvement of recognition accuracy on collected datasets. In this study, we focused on improving the recognition accuracy of air-writing data [4,27,28] collected from the LMC by combining representative features. The most successful classifier for spatiotemporal data, an HMM [36], was applied to features extracted from datasets D_1 [28] and D_2 [27] for 3D air-written word classification. The writing direction (wdir), curvature [34,37,38], and 3D point clouds establishing the trajectory (tr) [27] are common features for online handwriting analysis. In addition, the orthocenters (OC) [39] and ellipsoids [40] have recently been used to recognize 3D geometries.

We investigated the accuracy of word recognition using trajectories from datasets D_1 and D_2 considering various combinations of five features, $S = \{\text{wdir, curvature, tr, OC, ellipsoid}\}$, and HMM parameter settings. We aimed to find the best feature subset of S and HMM parameters that maximize the word recognition accuracy. Determining the best subset is difficult because there are $2^{|S|} - 1$ subsets, where $|S|$ is the set cardinality. Thus, a genetic algorithm (GA) was combined with an HMM classifier to obtain the best feature subset. Tenfold cross-validation was used to compute the accuracy, which was also analyzed according to the selection of the HMM parameters.

The remainder of this paper is organized as follows. Section 2 provides the theory of preprocessing, feature

extraction, HMM, GA, and integration of GA with HMM to select the best features. Experiments, results, and analyses are reported in Section 3 followed by a discussion in Section 4. Finally, conclusions are drawn in Section 5.

2 | METHODS

Five features, namely, wdir, curvature, tr, OC, and ellipsoid with dimensions 3, 11, 3, 3, and 24, respectively, were extracted from the aligned (preprocessed) sample trajectories of datasets D_1 and D_2 . A GA was applied for feature selection, and the selected features were fed into the HMM for word recognition. In this section, we detail each phase of the air-writing recognition method.

2.1 | Trajectory alignment

Preprocessing techniques, such as uniform sampling rate and alignment, help reduce errors owing to the quality of trajectory data [27,28]. Different sampling rates [27,28], writing styles across persons [27], and word alignments by a person at different times or across persons [27] are the main factors that determine the data quality. Trajectory alignment is the main quality issue in datasets D_1 and D_2 . We use the technique reported in [27,28] for trajectory alignment, which is an extension of slant correction for 2D words [34,41]. This technique combines regression [42,43], translation [44], and rotation [44].

Regression is used to determine the mainline of the trajectory. Orthogonal regression and principal component analysis (as available at <https://in.mathworks.com/help/stats/fitting-an-orthogonal-regression-using-principal-components-analysis.html>) are used to obtain the regression line, where the line direction is provided by the eigenvector corresponding to the largest eigenvalue of the covariance matrix of the trajectory tr . Let $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ be the initial point and an arbitrary fixed point (other than the initial point) on the regression line, respectively. The aligned trajectory (Atr) is obtained as $\mathbf{T}_3 * \mathbf{T}_2 * \mathbf{T}_1(tr)$, where

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_2 = \begin{bmatrix} u & v & 0 & 0 \\ \frac{u}{\sqrt{u^2+v^2}} & \frac{v}{\sqrt{u^2+v^2}} & 0 & 0 \\ -v & u & 0 & 0 \\ \frac{-v}{\sqrt{u^2+v^2}} & \frac{u}{\sqrt{u^2+v^2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_3 = \begin{bmatrix} \frac{w}{\sqrt{u^2+v^2+w^2}} & 0 & \frac{-\sqrt{u^2+v^2}}{\sqrt{u^2+v^2+w^2}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{u^2+v^2}}{\sqrt{u^2+v^2+w^2}} & 0 & \frac{w}{\sqrt{u^2+v^2+w^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $(x_2 - x_1, y_2 - y_1, z_2 - z_1) = (u, v, w)$. \mathbf{T}_1 is a translation matrix [44], while \mathbf{T}_2 and \mathbf{T}_3 are rotation matrices [44], and matrix product $\mathbf{T}_3 * \mathbf{T}_2 * \mathbf{T}_1$ provides the aligned trajectories along the Z axis.

2.2 | Feature extraction

Features $wdir$ and curvature provide angular and geometrical information, respectively, of 2D handwriting samples [27,34,37]. Kumar and others [27,28] extended the formulas of $wdir$ and curvature to air-writing samples. Let $P(x(t-1), y(t-1), z(t-1)) = P(p_x, p_y, p_z)$, $Q(x(t), y(t), z(t)) = Q(q_x, q_y, q_z)$, and $R(x(t+1), y(t+1), z(t+1)) = R(r_x, r_y, r_z)$ be three neighboring points along a trajectory. The direction cosines of the tangent to curve PQR define $wdir$ (α, β, γ) with three dimensions at point Q , where

$$\text{cosine}(\alpha) = \frac{d_x}{|d|}, \text{cosine}(\beta) = \frac{d_y}{|d|}, \text{cosine}(\gamma) = \frac{d_z}{|d|},$$

$$\vec{d} = \langle r_x - p_x, r_y - p_y, r_z - p_z \rangle = \langle d_x, d_y, d_z \rangle, \quad (1)$$

$$|d| = \sqrt{d_x^2 + d_y^2 + d_z^2}.$$

The curvature of segment PQR at point Q is determined by five subfeatures that are defined by a unique circumcircle of non-collinear points P, Q , and R , such that the curvature is on plane Δ defined by those points. Center (C) has dimension 3, radius η has dimension 1 of the circumcircle, two vectors \vec{CM} and \vec{CN} have dimension 3, where M and N are the midpoints of \vec{PQ} and \vec{QR} , respectively, and the exterior of $\angle PCR$ has dimension 1. Overall, the curvature has dimension 11. Note that (i) $\angle PCR$ is measured in radians and has a value above π rad; (ii) \vec{CM} and \vec{CN} are bisectors perpendicular to \vec{PQ} and \vec{QR} , respectively; and (iii) C is defined as

$$\vec{OC} = \frac{\sin(2\angle P)\vec{OP} + \sin(2\angle Q)\vec{OQ} + \sin(2\angle R)\vec{OR}}{\sin(2\angle P) + \sin(2\angle Q) + \sin(2\angle R)}, \quad (2)$$

where $\angle RPQ = \angle P$, $\angle PQR = \angle Q$, and $\angle PRQ = \angle R$, which are measured in radians and have values below π rad, and O is the trajectory origin.

Feature tr is determined by 3D point clouds [27] given by the position vectors of the trajectory corresponding to the movement of the finger along the X, Y , and Z axes, that is, $(x(t), y(t), z(t))$. The dimension of tr is 3.

Remondino [39] used OC to reconstruct 3D images of the human body. OC A of the triangle constructed by points P, Q , and R is computed as follows:

$$\vec{OA} = \frac{\tan(\angle P)\vec{OP} + \tan(\angle Q)\vec{OQ} + \tan(\angle R)\vec{OR}}{\tan(\angle P) + \tan(\angle Q) + \tan(\angle R)}. \quad (3)$$

Malyugina and others [40] considered an ellipsoid for the magnetometer calibration. The ellipsoid is a quadric surface area that can be represented as

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx + 2Hy + 2Iz + J = 0. \quad (4)$$

Most ellipsoid fitting methods are based on the least-squares method [45], [46], [47], where the fitness function is defined in terms of (i) geometric error, (ii) algebraic error, or (iii) approximate mean squared distance [48]. Methods using algebraic errors achieve efficient computation but lack geometric understanding [46]. Nevertheless, in [40], the algebraic method of [47] was shown to outperform other methods. Hence, we fit ellipsoids for 16 cloud points of word sequences using the algebraic method [47] subject to constraint $A + B + C = 3$. Therefore, nine ellipsoid parameters were estimated and used to estimate the center with dimension 3, radii with dimension 3, and radius directions with dimension 9. The resulting ellipsoid feature has dimension 24.

2.3 | HMM

An HMM is a sequential state machine used for modeling a wide range of time-series data [18,32,41,49,50]. In this section, we describe the general architecture of an HMM, its training algorithm, and the algorithm for classifying word labels.

The architecture of an HMM depends on the type of observation or feature data. If the features are discrete, the HMM is described by N hidden states $o_t \in \{1, 2, \dots, N\}$ at time t , M visible states

$v_t \in \{1, 2, \dots, M\}$ at time t , $N \times N$ transition probabilities a_{ij} (probability of moving from hidden state i at time $t - \omega_t = i$ to hidden state j at time $t + 1 - \omega_{t+1} = j$), $N \times M$ emission probabilities b_{ik} (probability of observing visible state k at time $t - v_t = k$ when the hidden state of the HMM at time t is i ($\omega_t = i$)), and N initial state probabilities π_i of ω_1 being i . If the features are continuous, the emission probabilities are replaced by the observation probabilities. Observation probabilities $b_i(x)$ of the features being x for hidden state i ($\omega_t = i$ at time t) are modeled as follows:

$$b_i(x) = \sum_{m=1}^M c_{im} \Pi(x, \mu_{im}, \sqcup_{im}), \quad i = 1, 2, \dots, N, \quad (5)$$

where x is the modeled vector of dimension l , c_{im} is the coefficient for mixture m in state $\omega_t = i$ with constraints

$$\sum_{m=1}^M c_{im} = 1, \quad i = 1, 2, \dots, N, \quad (6)$$

$$c_{im} \geq 0, \quad i = 1, 2, \dots, N, \quad m = 1, 2, \dots, M, \quad (7)$$

Π is any log-concave or elliptically symmetric probability density function, μ_{im} is the mean vector of dimension l and \sqcup_{im} is the covariance matrix of dimension $l \times l$ of mixture m in state $\omega_t = i$. If $X = (x_1, x_2, \dots, x_T)$ is an observation sequence, $\lambda = (\{a_{ij}\}_{ij}, \{b_i(x)\}_i, \{\pi_i\}_i)$ is given by the HMM, and $\Omega = (\omega_1, \omega_2, \dots, \omega_T)$ is a sequence of hidden states of λ . Then, the likelihood of observation sequence X is computed as follows:

$$P(X|\lambda) = \sum_{\Omega} \pi_{\omega_1} b_{\omega_1}(x_1) \prod_{t=2}^T a_{\omega_{t-1}\omega_t} b_{\omega_t}(x_t). \quad (8)$$

An alternative to computing the likelihood of observation sequence X is given by

$$P(X|\lambda) = \max_{\Omega} \pi_{\omega_1} b_{\omega_1}(x_1) \prod_{t=2}^T a_{\omega_{t-1}\omega_t} b_{\omega_t}(x_t). \quad (9)$$

Because we consider continuous time-series features, a continuous-density HMM is adopted. A Gaussian density is considered for each mixture, establishing a Gaussian mixture model. Different HMMs are trained for different words/labels. Parameters $(a_{ij}, c_{im}, \mu_{im}, \sqcup_{im}, \pi_i)$ of the HMM (λ_w) for word w are estimated for different values of N and M such that

likelihood $P(X|\lambda_w)$ is maximized with respect to the parameters. Parameter estimation comprises three main steps: (i) Initialization, (ii) re-estimation, and (iii) termination. Training of HMM λ_w given N , M , and X (X is a feature sequence corresponding to the word w) can be summarized as follows:

1. Use uniform segmentation for parameter initialization.
2. Use Viterbi algorithm [36,50] to estimate the best state sequence, $\Omega = (\omega_1, \omega_2, \dots, \omega_T)$. This sequence helps to calculate likelihood $P(X|\lambda_w)$.
3. Use Baum-Welch method [36,50] for re-estimation of model parameters. Baum-Welch re-estimation is formulated by maximizing Baum's auxiliary function [36]. The re-estimation formulae for a_{ij} , π_i , c_{im} , μ_{im} , and \sqcup_{im} are given by

$$\bar{\pi}_i = \text{expected number of times in state } i \text{ at time } t = 1, \quad (10)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to } j}{\text{expected number of transitions from state } i}, \quad (11)$$

$$\bar{c}_{im} = \frac{\text{expected number of times the system is in state } i \text{ using the } m\text{th mixture component}}{\text{expected number of times the system is in state } i}, \quad (12)$$

$$= \frac{\sum_{t=1}^T \gamma_t(i, m)}{\sum_{m=1}^M \sum_{t=1}^T \gamma_t(i, m)}, \quad (12)$$

$$\bar{\mu}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) x_t}{\sum_{t=1}^T \gamma_t(i, m)}, \quad (13)$$

$$\bar{\sqcup}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) (x_t - \mu_{im})(x_t - \mu_{im})'}{\sum_{t=1}^T \gamma_t(i, m)}, \quad (14)$$

where $\gamma_t(i, m)$ is the probability that x_t has been generated by the m -th mixture component in state i at time t .

4. Steps 2 and 3 are repeated until convergence of likelihood $P(X|\lambda)$.

Our training dataset contains many examples of a given word w . Therefore, the estimated HMM (λ_w) is updated using all examples of word w in the training dataset. This is applied to train all the HMMs. An unknown feature sequence X is recognized by computing the likelihood of all the HMMs. Sequence X is recognized as w^* as follows [36,50]:

$$w^* = \operatorname{argmax}_{w \in W} \{P(X|\lambda_w)\}. \quad (15)$$

2.4 | Feature selection

Every feature contributes to the computational cost and accuracy of a recognition system. If we select few features, the computational cost decreases at the expense of accuracy. Therefore, an optimal subset must be selected from a set of feasible features [51]. Five feasible features are used in our method. The possible feature subsets are $S_1 = \{\text{wdir}\}$, $S_2 = \{\text{OC}\}$, $S_3 = \{\text{tr}\}$, $S_4 = \{\text{curvature}\}$, $S_5 = \{\text{ellipsoid}\}$, $S_6 = \{\text{OC, wdir}\}$, $S_7 = \{\text{tr, wdir}\}$, $S_8 = \{\text{tr, OC}\}$, $S_9 = \{\text{curvature, wdir}\}$, $S_{10} = \{\text{tr, curvature}\}$, $S_{11} = \{\text{OC, curvature}\}$, $S_{12} = \{\text{wdir, ellipsoid}\}$, $S_{13} = \{\text{tr, ellipsoid}\}$, $S_{14} = \{\text{OC, ellipsoid}\}$, $S_{15} = \{\text{curvature, ellipsoid}\}$, $S_{16} = \{\text{tr, OC, wdir}\}$, $S_{17} = \{\text{tr, OC, curvature}\}$, $S_{18} = \{\text{curvature, wdir, OC}\}$, $S_{19} = \{\text{tr, wdir, curvature}\}$, $S_{20} = \{\text{OC, wdir, ellipsoid}\}$, $S_{21} = \{\text{ellipsoid, tr, wdir}\}$, $S_{22} = \{\text{tr, OC, ellipsoid}\}$, $S_{23} = \{\text{tr, curvature, ellipsoid}\}$, $S_{24} = \{\text{wdir, curvature, ellipsoid}\}$, $S_{25} = \{\text{OC, curvature, ellipsoid}\}$, $S_{26} = \{\text{tr, OC, wdir, curvature}\}$, $S_{27} = \{\text{OC, wdir, tr, ellipsoid}\}$, $S_{28} = \{\text{tr, OC, wdir, ellipsoid}\}$, $S_{29} = \{\text{tr, ellipsoid, curvature, wdir}\}$, $S_{30} = \{\text{OC, wdir, curvature, ellipsoid}\}$, and $S_{31} = \{\text{tr, OC, wdir, curvature, ellipsoid}\}$.

We aimed to select the best combination of features for air-writing recognition. This combination was obtained using a GA [51,52]. The use of a GA for feature selection is described in Algorithm 1. The GA was applied to 35 and 10 randomly selected samples (D' , M , N) from datasets D_1 and D_2 , respectively. Algorithm 1 selected feature combinations S_{16} , S_{26} , and S_{31} for D_1 with frequencies of 10, 16, and 9, respectively, and S_{16} and S_{26} for D_2 with frequencies of 1 and 9, respectively.

3 | EXPERIMENTS, RESULTS, AND ANALYSES

The HMMs were constructed for the recognition of air-written words from samples in datasets D_1 and D_2 . The selected sets of features (S_{16} , S_{26} , S_{31}) by the GA were used in the HMMs. The 10-fold cross-validation accuracy of the recognized words was used to evaluate the performance of each HMM for the datasets and feature subsets (D_1 , S_{16} , D_1 , S_{26} , D_1 , S_{31} , and D_2 , S_{26}). Cross-validation was performed on HMM parameters $N \in \{1, 2, 3, 4\}$ and $M \in \{2, 4, 8, \dots, 256\}$.

The 10-fold cross-validation can be summarized as follows. Divide the given dataset into 10 disjoint subsets. Use nine subsets for training, and the remaining subset

for validation. Repeat this process until exploring all the combinations of the 10 disjoint subsets. All the combinations of N and M with the 10 disjoint subsets provided $4 \times 8 \times 10 = 320$ validation accuracy values. We used M – accuracy, M – opt, N – accuracy, and N – opt given by

$$M\text{ – accuracy} = \max_M \{\text{accuracy}\}, \quad (16)$$

$$M\text{ – opt} = \operatorname{argmax}_M \{\text{accuracy}\}, \quad (17)$$

$$N\text{ – accuracy} = \max_N \{\text{accuracy}\}, \quad (18)$$

$$N\text{ – opt} = \operatorname{argmax}_N \{\text{accuracy}\} \quad (19)$$

to search for the values of M and N providing the highest accuracy. All the experiments were implemented on the 64-bit MathWorks MATLAB R2016b software.

The values of M – accuracy and M – opt and those of N – accuracy and N – opt are listed in Tables 1 and 2, respectively, for datasets D_1 and D_2 . The best values for (M – accuracy, N – accuracy) are (97.62%, 972.6%), (98.81%, 98.81%), (86.9%, 86.9%), and (83.58%, 83.58%) for (D_1 , S_{16}), (D_1 , S_{26}), (D_1 , S_{31}), and (D_2 , S_{26}), respectively. The corresponding values for (M – opt, N – opt) are (128, 4), (128, 2), (256, 3), and (64, 4). The best accuracy for D_1 of 98.81% corresponds to feature subset S_{26} . A detailed analysis of D_1 with S_{26} is shown as a confusion matrix in Figure 1 for $(M, N) = (128, 2)$. Word max is the only misclassified class, being incorrectly recognized as word navy, possibly owing to the misclassification of letter m as n and x as vy.

3.1 | Error analysis on (D_1, S_{26}) for (M, N) of (64, 2) and (128, 2)

Table 3 depicts three trajectories of dataset D_1 that were incorrectly recognized when using HMM parameters $(M, N) = (64, 2)$ but correctly recognized when using $(M, N) = (128, 2)$. (M, N) of (64, 2) corresponds to the second-best recognition accuracy (97.62% in Tables 1 and 2). In the first row of Table 3, the word heap is incorrectly recognized as the word has. Letters ea are recognized as letter a, and letter p is recognized as letter s owing to their similar trajectories. In the second row, the word conquer is wrongly recognized as the word dress, owing to confusion in the writing style in a single stroke. In [28], the word access was incorrectly recognized as the word dress. However, in our method, the words conquer

TABLE 1 Values of (M – accuracy, M – opt) in 10-fold cross-validation

N	1	2	3	4
Data tuple: (D_1, S_{16})				
10 th fold	96.43, 256	95.24, 128	92.86, {8, 32, 64}	92.86, {8, 32, 64}
9 th fold	94.05, 128	96.43, 32	91.67, {8, 16, 256}	91.67, {8, 16, 256}
8 th fold	96.43, 256	95.24, {64, 128}	96.43, 256	94.05, 32
7 th fold	90.48, 64	88.10, {32, 64, 256}	89.29, {64, 128}	89.29, {64, 128}
6 th fold	95.24, {64, 128}	96.43, {128, 256}	95.24, {64, 128}	97.62, 128
5 th fold	90.48, 256	85.71, 64	89.29, 32	88.10, {64, 128}
4 th fold	92.86, {64, 128}	95.24, 64	91.67, {8, 32, 64}	94.05, 256
3 rd fold	94.05, 64	92.86, 128	91.67, {64, 256}	91.67, {64, 256}
2 nd fold	92.86, {32, 64, 128, 256}	92.86, {32, 64, 128, 256}	92.86, {32, 64, 128, 256}	92.67, 32
1 st fold	85.71, 128	84.52, {64, 128, 256}	84.52, {64, 128, 256}	84.52, {64, 128, 256}
Data tuple: (D_1, S_{26})				
10 th fold	95.24, {32, 64, 128, 256}	95.24, {32, 64, 128, 256}	94.05, 64	95.24, {32, 64, 128, 256}
9 th fold	94.05, {32, 64, 256}	97.62, 64	94.05, {32, 64, 256}	96.43, 32
8 th fold	95.24, 64	98.81, 128	96.43, {32, 128, 256}	96.43, {32, 128, 256}
7 th fold	92.86, {32, 64, 256}	92.86, {32, 64, 256}	91.67, 128	91.67, 128
6 th fold	97.62, {64, 128, 256}	97.62, {64, 128, 256}	96.43, {64, 128}	96.43, {64, 128}
5 th fold	90.48, {64, 128}	91.67, 256	89.29, {32, 128}	90.48, {64, 128}
4 th fold	94.05, 64	96.43, {64, 128}	92.86, {16, 32}	91.67, 128
3 rd fold	92.50, 32	96.43, 128	90.10, 64	88.20, 32
2 nd fold	94.05, 64	96.43, {64, 128}	94.05, 64	93.10, 128
1 st fold	88.38, 64	94.34, 128	95.7, 128	89.29, 32
Data tuple: (D_1, S_{31})				
10 th fold	78.57, 256	77.38, {64, 128}	77.38, {64, 128}	78.9, 256
9 th fold	82.05, 256	82.86, 256	75.40, 32	77.62, 64
8 th fold	86.43, 128	85.04, 256	86.9, 128	81.67, 256
7 th fold	76.19, 64	84.05, 64	85.4, 128	86.43, 256
6 th fold	75.24, 128	76.43, 128	85.24, 64	86.62, {128, 256}
5 th fold	80.48, 128	85.70, 64	83.29, 64	81.67, {128, 256}
4 th fold	72.86, 64	75.24, 128	83.37, 64	86.43, {64, 128, 256}
3 rd fold	84.05, 64	82.86, 128	81.67, 64	86.43, 64
2 nd fold	82.86, {64, 128}	79.86, 64	76.43, 64	85.86, 256
1 st fold	85.71, 128	84.52, {64, 256}	86.90, 256	84.52, {64, 256}
Data tuple: (D_2, S_{26})				
10 th fold	68.57, 256	67.38, 128	73.38, 64	79.90, 256
9 th fold	66.05, 256	73.33, 128	75.67, 16	83.58, 64
8 th fold	66.43, 128	65.04, 256	66.90, 128	76.60, 64
7 th fold	66.19, 64	67.10, 64	76.43, 256	80.05, 64
6 th fold	66.24, 128	71.43, 128	70.24, 64	82.62, 128
5 th fold	79.48, 128	78.70, 64	73.29, 32	81.58, 256

(Continues)

TABLE 1 (Continued)

N	1	2	3	4
4 th fold	72.86, 64	75.24, 128	81.23, 64	82.43, 64
3 rd fold	74.05, 64	72.86, 128	81.07, 64	83.43, 128
2 nd fold	72.86, 128	79.86, 64	76.43, 64	80.86, 256
1 st fold	65.71, 128	64.52, 64	74.52, 64	71.20, 8

Note: Bold values correspond maximum M-accuracy.

TABLE 2 Values of (N – accuracy, N – opt) in 10-fold cross-validation

M	2	4	8	16	32	64	128	256
Data tuple: (D ₁ , S ₁₆)								
10 th fold	70.24, 4	85.71, 4	92.86, {1, 3, 4}	90.48, 2	92.86, {1, 3, 4}	94.05, 1	95.24, 2	96.43, 1
9 th fold	57.14, 4	84.52, 4	91.67, {3, 4}	91.67, {3, 4}	96.43, 2	95.24, 2	94.05, 1	91.67, {3, 4}
8 th fold	46.43, 4	73.81, 4	89.29, 4	94.05, 3	94.05, 4	95.24, 2	95.24, 2	96.43, {1, 3}
7 th fold	54.76, 3	73.81, 4	82.14, 4	86.90, 3	88.10, {1, 2, 3}	90.48, 1	89.29, {3, 4}	88.1, {1, 2, 3}
6 th fold	53.57, 4	77.38, 4	85.71, 2	91.67, 2	94.05, 2	95.24, {1, 3}	97.62, 4	96.43, 2
5 th fold	54.76, 4	75.00, 3	79.76, 3	84.82, 2	89.29, 3	88.10, 4	88.10, 4	90.48, 1
4 th fold	60.71, 4	80.95, 3	91.67, {1, 3, 4}	90.48, 1	91.67, {1, 3, 4}	95.24, 2	92.86, {1, 2}	94.05, 4
3 rd fold	59.52, 4	73.81, 4	82.14, 3	88.10, 4	90.48, 4	94.05, 1	92.86, 2	91.67, {2, 3, 4}
2 nd fold	45.24, 4	75.00, 4	86.90, 4	91.67, 3	92.86, {1, 2, 3}	92.86, {1, 2, 3}	92.86, {1, 2, 3}	92.86, {1, 2, 3}
1 st fold	48.81, 4	67.86, 4	72.62, 2	79.76, 3	83.33, 3	84.52, {2, 4}	85.71, 1	84.52, {2, 4}
Data tuple: (D ₁ , S ₂₆)								
10 th fold	55.95, 4	85.71, 4	84.52, 2	90.48, 4	95.24, {1, 2, 4}	95.24, {1, 2, 4}	95.24, {1, 2, 4}	95.24, {1, 2, 4}
9 th fold	57.14, 4	82.14, 4	88.10, 4	95.24, 4	96.43, 2	97.62, 2	95.24, 2	94.05, {1, 3}
8 th fold	45.24, 4	72.62, 4	85.71, 2	91.67, 4	96.43, {2, 3, 4}	97.62, 2	98.81, 2	96.43, {2, 3, 4}
7 th fold	46.43, 4	75.00, 4	86.90, 4	88.10, 2	92.86, {1, 2}	92.86, {1, 2}	91.67, {1, 3, 4}	92.86, {1, 2}
6 th fold	53.57, 4	76.19, 4	86.90, 4	89.29, 4	91.67, 2	97.62, {1, 2}	97.62, {1, 2}	97.62, {1, 2}
5 th fold	48.81, 3	73.81, 4	83.33, 4	86.90, 4	89.29, {2, 3}	90.48, {1, 2, 4}	90.48, {1, 2, 4}	91.67, 2
4 th fold	45.67, 4	75.00, 4	90.48, 4	92.86, {2, 3}	95.24, 2	96.43, 2	96.43, 2	95.24, 2
3 rd fold	46.43, 4	72.62, 3	82.14, 4	86.90, 3	95.24, 2	91.67, 1	96.43, 2	91.67, 1
2 nd fold	46.43, 3	72.62, 4	82.14, 4	86.90, 3	91.67, 4	96.43, 2	96.43, 2	92.86, 1
1 st fold	45.24, 4	67.86, 3	83.33, 3	88.01, 4	89.29, 4	90.48, 3	95.7, 3	88.10, 2
Data tuple: (D ₁ , S ₃₁)								
10 th fold	24.50, 1	28.05, 4	55.09, 3	61.90, 3	70.8, 4	77.38, {2, 3, 4}	77.38, {2, 3, 4}	78.90, 4
9 th fold	26.9, 2	28.90, 3	70.24, 4	68.20, 2	76.40, 4	77.62, 4	75.24, 4	82.86, 2
8 th fold	27.9, 1	27.90, 2	73.20, 3	70.2, 3	75.4, 4	80.48, 4	86.90, 3	85.04, 2
7 th fold	28.90, 2	29.00, 4	71.43, 4	68.9, 4	76.90, 4	84.05, 2	85.40, 3	86.43, 4
6 th fold	29.76, 2	27.9, 2	72.62, 3	69.90, 3	75.80, 4	86.29, 4	86.62, 4	86.62, 4
5 th fold	30.1, 4	32.10, 3	76.30, 4	75.80, 3	73.6, 4	85.70, 2	81.67, 4	81.67, 4
4 th fold	31.20, 2	34.2, 3	72.4, 4	69.80, 4	72.60, 4	86.43, 4	86.43, 4	85.24, 4
3 rd fold	34.5, 2	35.7, 1	75.30, 3	73.70, 4	79.30, 4	86.43, 4	84.43, 4	82.86, 4
2 nd fold	35.7, 1	38.7, 2	74.30, 3	78.9, 4	80.40, 4	82.86, {1, 4}	84.05, 4	85.86, 4
1 st fold	30.1, 2	39.9, 3	75.0, 4	76.4, 4	84.8, 3	85.04, 3	85.71, 1	86.90, 3

(Continues)

TABLE 2 (Continued)

M	2	4	8	16	32	64	128	256
Data tuple: (D_2, S_{26})								
10 th fold	29.50, 1	28.05, 3	70.09, 4	61.90, 4	72.34, 4	73.38, 3	77.40, 4	79.90, 4
9 th fold	37.90, 3	38.9, 2	74.24, 4	75.67, 3	56.40, 4	83.58, 4	80.24, 4	66.05, 1
8 th fold	36.90, 2	29.90, 1	50.20, 2	54.20, 3	64.9, 3	76.60, 4	66.90, 3	65.04, 2
7 th fold	27.90, 1	32.00, 3	61.430, 3	67.90, 4	63.90, 3	80.05, 4	75.40, 3	76.43, 3
6 th fold	28.76, 1	37.90, 1	62.62, 3	69.70, 3	70.70, 4	81.29, 4	82.62, 4	63.62, 3
5 th fold	30.10, 1	35.10, 1	67.30, 4	64.8, 3	66.60, 4	80.08, 4	79.48, 1	81.58, 4
4 th fold	32.2, 1	44.20, 3	62.40, 4	70.80, 4	71.60, 4	82.43, 4	76.43, 4	75.24, 4
3 rd fold	39.50, 1	38.70, 1	65.30, 3	72.7, 4	72.8, 4	81.30, 4	83.43, 4	82.34, 4
2 nd fold	38.70, 1	42.70, 2	64.30, 3	68.90, 4	71.40, 4	79.86, 2	76.34, 3	80.86, 4
1 st fold	39.10, 1	36.90, 2	71.20, 4	66.40, 4	64.80, 4	74.52, 3	71.67, 3	66.25, 4

Note: Bold values correspond maximum N-accuracy.

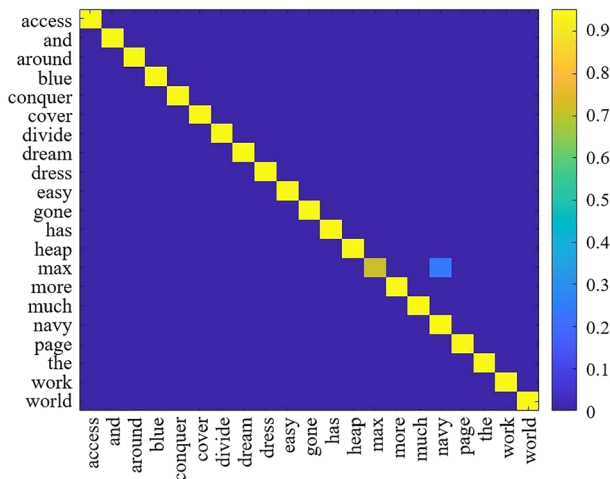


FIGURE 1 Confusion matrix of air-writing recognition accuracy of words. The predicted and true labels are listed on the horizontal and vertical axes, respectively.

and access are correctly recognized when using $(M, N) = (128, 2)$. The word much in the third row is recognized as the word max because letter u is recognized as letter a and the last two letters, ch, are recognized as letter x.

4 | DISCUSSION

The recognition accuracies of the air-written words obtained from the proposed and existing methods [26], [27], and [28] for datasets D_1 and D_2 are listed in Table 4. In all the compared studies, HMM classifiers were used.

TABLE 3 Samples from dataset D_1 incorrectly recognized when using HMM parameters $(M, N) = (64, 2)$. All these samples are correctly recognized when using $(M, N) = (128, 2)$

Original Trajectory	Recognized Trajectory by $(M, N) = (64, 2)$	Original Word	Recognized Word by $(M, N) = (64, 2)$
		heap	has
		conquer	dress
		much	max

Fast processing with high accuracy on low-memory devices is the goal of air-writing recognition. The accuracy on datasets D_1 and D_2 is improved using the proposed method. This improvement is due to the better feature selection (S_{26}) provided by the GA (Algorithm 1) from a large set of feasible feature combinations. Both datasets benefit from set S_{26} in terms of accuracy. Because Algorithm 1 is not used in other methods, the computational and memory costs of our method are higher. However, the accuracy improvement achieved by the GA justifies its use. Furthermore, the computational cost of the HMM using S_{26} is higher than that using S_{19} because S_{26} contains more features. Regarding large-scale feature issues, dimensionality reduction may help balance the tradeoff between accuracy and computational cost.

TABLE 4 Comparison between proposed and existing methods for air-writing recognition on datasets D_1 and D_2

Study	Dataset	Features	Accuracy
This study	D_1	tr, OC, wdir, curvature (S_{26})	98.81%
Kumar et al. (2017) [28]	D_1	tr, wdir, curvature (S_{19})	92.70%
Agarwal et al. (2015) [26]	D_1	word boundaries with help of partial differentiation	77.62%
This study	D_2	tr, OC, wdir, curvature (S_{26})	83.58%
Kumar et al. (2016) [27]	D_2	tr, wdir, curvature (S_{19})	81.25%

Over the past few years, deep-learning-based methods have provided outstanding results for classification problems, including sequence labeling. Recursive neural networks, RNNs, and convolutional neural networks (CNNs) are well-known deep networks [53]. The bidirectional LSTM architecture [53] of an RNN with connectionist temporal classification as its output layer is the most successful deep network for sequence labeling. The bidirectional LSTM combines LSTM [54] with a bidirectional RNN [55]. The bidirectional LSTM network stores information in the past and future (through the bidirectional RNN) of a specific timeframe using long- and short-term memory (from LSTM). On the other hand, the performance of an attention recurrent translator is comparable to that of connectionist temporal classification for air-written English word recognition. Deep CNNs and multi-column deep networks have provided promising results for offline handwriting recognition. The CNN-based DeepCNet won the first place in the ICDAR 2013 Chinese handwriting recognition competition, and its results have been improved by advanced CNN architectures including DropWeight, DropDistortion, and DropSample [56].

Kumar and others [27] obtained an accuracy of 86.88% for dataset D_2 by using S_{19} and a bidirectional LSTM network. The accuracy of our proposed method is 83.58% on dataset D_2 using S_{26} and the HMM. When the HMM is used, feature set S_{26} outperforms S_{19} in terms of accuracy. When S_{19} is used, the bidirectional LSTM network outperforms the HMM in terms of accuracy (Table 4). Hence, using feature set S_{26} in the bidirectional LSTM network may improve accuracy. However, the recognition accuracy of any sequence labeling method

depends on its architecture, and the optimal architecture for the bidirectional LSTM network must be determined to reach its maximum possible accuracy. In this study, experiments were not performed to optimize the architecture of the bidirectional LSTM network. Graham [57] used DeepCNet to label unidimensional temporal sequences. In our study, air-written words were stored in a 3D temporal sequence. Extending DeepCNet to 3D temporal sequence recognition may improve the recognition accuracy on datasets D_1 and D_2 .

5 | CONCLUSIONS

The recognition accuracy of air-written words captured by the LMC was studied with respect to various sets of features and parameters of an HMM classifier. Feature sets were obtained from the combinations of five features, namely, wdir, curvature, tr, OC, and ellipsoid. Good combinations of features obtained by a GA included S_{16} (tr, OC, wdir), S_{26} (tr, OC, wdir, curvature), and S_{31} (tr, OC, wdir, curvature, and ellipsoid). These feature sets were used by the HMM classifier for word recognition in samples from datasets D_1 and D_2 . Feature set S_{26} provided the best recognition on both datasets. In addition, the best HMM parameter pairs (M, N) for datasets D_1 and D_2 were (128, 2) and (64, 4), respectively. Dataset D_1 led to the highest recognition accuracy of 98.81%, while an accuracy of 83.58% was obtained for dataset D_2 , which was slightly lower than the highest accuracy of 86.88% obtained in [27]. The highest accuracy for dataset D_2 was achieved using a bidirectional LSTM network.

In future work, we will combine HMMs and deep networks to improve the accuracy of air-writing recognition. The computational and memory costs were higher in our method than in previous studies that neglected feature selection. We verified an improvement in accuracy using feature selection, which is thus necessary. Furthermore, because S_{26} had a larger dimension than S_{19} , the computational cost of an HMM using S_{26} was higher. To handle large-scale feature issues, dimensionality reduction should be explored to balance the tradeoff between the accuracy and computational cost. Eventually, we intend to develop a framework suitable for making notes collected from air-written words.

AUTHOR CONTRIBUTION

Deval Verma conducted this work during her PhD. studies at the Jaypee Institute of Information Technology. Dr. Amrish Kumar Aggrawal and Dr. Himanshu Agarwal were the PhD supervisors.


ACKNOWLEDGMENTS

We thank Dr. Partha Pratim Roy and Dr. Rajkumar Saini from the Department of Computer Science and Engineering, Indian Institute of Technology (IIT), Roorkee, India, for their help with the data collection and valuable suggestions. The authors also acknowledge the research support provided by the Jaypee Institute of Information Technology, Noida, India.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest.

ORCID

Himanshu Agarwal  <https://orcid.org/0000-0002-9950-7447>

REFERENCES

- G. D'Amico, A. Del Bimbo, F. Dini, L. Landucci, and N. Torpei, *Natural human-computer interaction*, Multimedia interaction and intelligent user interfaces, Springer, 2010, pp. 85–106.
- M. Alam, K.-C. Kwon, M. Y. Abbass, S. M. Imtiaz, and N. Kim, *Trajectory-based air-writing recognition using deep neural network and depth sensor*, *Sensors* **20** (2020), no. 2, 376.
- D. H. Kim, H. I. Choi, and J. H. Kim, 3D space handwriting recognition with ligature model, *International symposium on ubiquitous computing systems*, Springer, 2006, pp. 41–56.
- 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-Mach. Syst.* **46** (2015), no. 3, 403–413.
- P. P. Roy, A. K. Bhunia, A. Das, P. Dey, and U. Pal, *HMM-based Indic handwritten word recognition using zone segmentation*, *Pattern Recog.* **60** (2016), 1057–1075.
- R. Plamondon and S. N. Srihari, *Online and off-line handwriting recognition: A comprehensive survey*, *IEEE Trans. Pattern Anal. Mach. Intell.* **22** (2000), no. 1, 63–84.
- O. Patsadu, C. Nukoolkit, and B. Watanapa, *Human gesture recognition using Kinect camera*, (Ninth International Conference on Computer Science and Software Engineering, Bangkok, Thailand), 2012, pp. 28–32.
- S. Ghosh, S. Ghosh, P. Kumar, E. Scheme, and P. P. Roy, *A novel spatio-temporal siamese network for 3D signature recognition*, *Pattern Recog. Lett.* **144** (2021), 13–20.
- E. Guerra-Segura, A. Ortega-Pérez, and C. M. Travieso, *In-air signature verification system using leap motion*, *Expert Syst. Appl.* **165** (2021), 113797.
- J. Han, L. Shao, D. Xu, and J. Shotton, *Enhanced computer vision with microsoft Kinect sensor: A review*, *IEEE Trans. Cybern.* **43** (2013), no. 5, 1318–1334.
- F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, *Analysis of the accuracy and robustness of the leap motion controller*, *Sensors* **13** (2013), no. 5, 6380–6393.
- S. Patil, D. Kim, S. Park, and Y. Chai, *Handwriting recognition in free space using WIMU-based hand motion analysis*, *J. Sensors* **2016** (2016). <https://doi.org/10.1155/2016/3692876>
- M. Arsalan and A. Santra, *Character recognition in air-writing based on network of radars for human-machine interface*, *IEEE Sensors J.* **19** (2019), no. 19, 8855–8864.
- S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, *Fingertip detection and tracking for recognition of air-writing in videos*, *Expert Syst. Appl.* **136** (2019), 217–229.
- S. Mohammadi and R. Maleki, *Air-writing recognition system for Persian numbers with a novel classifier*, *The Visual Comput.* **36** (2020), no. 5, 1001–1015.
- T. Guzsvinecz, V. Szucs, and C. Sik-Lanyi, *Suitability of the Kinect sensor and leap motion controller: A literature review*, *Sensors* **19** (2019), no. 5, 1072.
- T. Yanay and E. Shmueli, *Air-writing recognition using smart-bands*, *Pervasive Mob. Comput.* **66** (2020), 101183.
- H.-S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang, *Hand gesture recognition using combined features of location, angle and velocity*, *Pattern Recog.* **34** (2001), no. 7, 1491–1501.
- Z. Feng, S. Xu, X. Zhang, L. Jin, Z. Ye, and W. Yang, *Real-time fingertip tracking and detection using Kinect depth sensor for a new writing-in-the air system*, (Proceedings of the 4th International Conference on Internet Multimedia Computing and Service, Wuhan, China), 2012, pp. 70–74.
- X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, *A new writing experience: Finger writing in the air using a Kinect sensor*, *IEEE MultiMedia* **20** (2013), no. 4, 85–93.
- T. Murata and J. Shin, *Hand gesture and character recognition based on Kinect sensor*, *Int. J. Distrib. Sensor Netw.* **10** (2014), no. 7, 278460.
- R. Aggarwal, S. Swetha, A. M. Namboodiri, J. Sivaswamy, and C. V. Jawahar, *Online handwriting recognition using depth sensors*, (13th International Conference on Document Analysis and Recognition, Tunis, Tunisia), 2015, pp. 1061–1065.
- C. Wang, C.-Y. Su, and C.-L. Lin, *A novel recognition system for digits writing in the air using coordinated path ordering*, (International Conference on Intelligent Informatics and Biomedical Sciences, Okinawa, Japan), 2015, pp. 244–249.
- S. Vikram, L. Li, and S. Russell, *Handwriting and gestures in the air, recognizing on the fly*, *Proceedings of the CHI, Paris, France*, **13**, 2013, pp. 1179–1184.
- N. Xu, W. Wang, and X. Qu, *On-line sample generation for in-air written chinese character recognition based on leap motion controller*, (Pacific Rim Conference on Multimedia, Gwangju, Republic of Korea), 2015, pp. 171–180.
- C. Agarwal, D. P. Dogra, R. Saini, and P. P. Roy, *Segmentation and recognition of text written in 3D using leap motion interface*, 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, 2015, pp. 539–543.
- P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, *Study of text segmentation and recognition using leap motion sensor*, *IEEE Sensors J.* **17** (2017), no. 5, 1293–1301.
- P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, *3D text segmentation and recognition using leap motion*, *Multimedia Tools Appl.* **76** (2017), no. 15, 16491–16510.
- S. Misra, J. Singha, and R. H. Laskar, *Vision-based hand gesture recognition of alphabets, numbers, arithmetic operators and ascii characters in order to develop a virtual text-entry interface system*, *Neural Comput. Appl.* **29** (2018), no. 8, 117–135.

30. J. K. Oh, S.-J. Cho, W.-C. Bang, W. Chang, E. Choi, J. Yang, J. Cho, and D. Y. Kim, *Inertial sensor based recognition of 3-d character gestures with an ensemble classifiers*, (Ninth international workshop on frontiers in handwriting recognition, Kokubunji, Japan), 2004, pp. 112–117.
31. J.-S. Wang, Y.-L. Hsu, and C.-L. Chu, *Online handwriting recognition using an accelerometer-based pen device*, 2nd international conference on advances in computer science and engineering (cse 2013). Atlantis Press, 2013.
32. C. Amma, M. Georgi, and T. Schultz, *Airwriting: A wearable handwriting recognition system*, *Pers. Ubiquitous Comput.* **18** (2014), no. 1, 191–203.
33. D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbone, *Multi-language online handwriting recognition*, *IEEE Trans. Pattern Anal. Machine Intell.* **39** (2016), no. 6, 1180–1194.
34. S. Jaeger, S. Manke, J. Reichert, and A. Waibel, *Online handwriting recognition: The NPen++ recognizer*, *Int. J. Document Anal. Recogn.* **3** (2001), no. 3, 169–180.
35. P. P. Roy, P. Kumar, S. Patidar, and R. Saini, *3D word spotting using leap motion sensor*, *Multimedia Tools Appl.* **80** (2021), no. 8, 11671–11689.
36. L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, *Proc. IEEE* **77** (1989), no. 2, 257–286.
37. I. Guyon, P. Albrecht, Y. Le Cun, J. Denker, and W. Hubbard, *Design of a neural network character recognizer for a touch terminal*, *Pattern Recogn.* **24** (1991), no. 2, 105–119.
38. V. Ghods, E. Kabir, and F. Razzazi, *Decision fusion of horizontal and vertical trajectories for recognition of online Farsi subwords*, *Eng. Appl. Artif. Intell.* **26** (2013), no. 1, 544–550.
39. F. Remondino, *3D reconstruction of static human body with a digital camera*, 2003. *Videometrics VII*, 5013.
40. A. Malyugina, K. Igudesman, and D. Chickrin, *Least-squares fitting of a three-dimensional ellipsoid to noisy data*, *Appl. Math. Sci.* **8** (2014), no. 149, 7409–7421.
41. A. Bharath and S. Madhvanath, *HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts*, *IEEE Trans. Pattern Anal. Mach. Intell.* **34** (2012), no. 4, 670–682.
42. J. O. Rawlings, S. G. Pantula, and D. A. Dickey, *Applied regression analysis: A research tool*, Springer Science & Business Media, 2001.
43. G. M. Furnival and R. W. Wilson, *Regressions by leaps and bounds*, *Technometrics* **16** (1974), no. 4, 499–511.
44. J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, J. Hughes, and E. Angel, *Computer graphics: Principles and practice*, 3rd ed., Addison-Wesley Professional, 1996.
45. G. Calafiore, *Approximation of n-dimensional data using spherical and ellipsoidal primitives*, *IEEE Trans. Syst., Man, Cybern.-Part A: Syst. Humans* **32** (2002), no. 2, 269–278.
46. J. Yu, S. R. Kulkarni, and H. V. Poor, *Robust ellipse and spheroid fitting*, *Pattern Recogn. Lett.* **33** (2012), no. 5, 492–499.
47. D. A. Turner, I. J. Anderson, J. C. Mason, and M. G. Cox, *An algorithm for fitting an ellipsoid to data*, National Physical Laboratory, UK, 1999.
48. G. Taubin, *Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation*, *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (1991), 1115–1138.
49. D. Dimov and I. Azmanov, *Experimental specifics of using HMM in isolated word speech recognition*, 2005.
50. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, and D. Povey, *The HTK book*, Cambridge Univers. Eng. Dep. **3** (2002), 175.
51. H. Vafaie and K. A. De Jong, *Genetic algorithms as a tool for feature selection in machine learning*, (Proceedings Fourth International Conference on Tools with Artificial Intelligence, Arlington, VA, USA), 1992, pp. 200–203.
52. J. Huang, Y. Cai, and X. Xu, *A hybrid genetic algorithm for feature selection wrapper based on mutual information*, *Pattern Recogn. Lett.* **28** (2007), no. 13, 1825–1844.
53. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, *Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions*, *J. Big Data* **8** (2021), no. 1, 1–74.
54. S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural Comput.* **9** (1997), no. 8, 1735–1780.
55. M. Schuster and K. K. Paliwal, *Bidirectional recurrent neural networks*, *IEEE Trans. Signal Process.* **45** (1997), no. 11, 2673–2681.
56. S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, *Fingertip detection and tracking for recognition of air-writing in videos*, *Expert Syst. Appl.* **136** (2019), 217–229.
57. B. Graham, *Sparse arrays of signatures for online character recognition*, arXiv preprint, 2013. <https://doi.org/10.48550/arXiv.1308.0371>

AUTHOR BIOGRAPHIES



Deval Verma is an Assistant Professor at the School of Computer Science and Engineering, Bennett University, Greater Noida, India. She obtained her PhD degree in Mathematics from the Jaypee Institute of Information Technology, Noida, India, in 2021 and MSc degree in Mathematics from HNB Garhwal University, India, in 2007. Her research interests include digital watermarking, palmprint matching, optical character recognition, and machine learning. She has published many research papers in reputed international journals and conferences. Dr. Verma has served as a TPC member, session chair, and keynote speaker in reputed conferences and is an active reviewer for many international journals.



Himanshu Agarwal is an Assistant Professor at the Department of Mathematics, Jaypee Institute of Information Technology, Noida, India. He received his PhD degree from the Department of Mathematics, Indian Institute of Technology (IIT)

Roorkee in 2015. He was a Visiting Research Student at the Applied Computer Science Department, University of Winnipeg, Canada, in 2012–2013 for 6 months. He received his MSc degree in Industrial Mathematics and Informatics from IIT Roorkee in 2009. He has published over 20 research papers in reputed international journals and conferences. His research areas of interest include image processing, computer vision, information visualization, information security, machine learning, statistical learning, and differential equations.



Amrish Kumar Aggarwal is a Professor at the Department of Mathematics, Jaypee Institute of Information Technology, Noida, India, since 2007. Professor Aggarwal has 30 years of teaching and research experience. He has published over

40 research papers in reputed international journals and conference proceedings. Prof. Aggarwal supervised the elaboration of six PhD and two MTech Dissertations. His research interests include continuum mechanics, optimization, and image processing.

How to cite this article: D. Verma, H. Agarwal, and A. K. Aggarwal, *Selection of features and hidden Markov model parameters for English word recognition from Leap Motion air-writing trajectories*, ETRI Journal **46** (2024), 250–262. DOI [10.4218/etrij.2022-0266](https://doi.org/10.4218/etrij.2022-0266)