IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 April 2023, accepted 19 April 2023, date of publication 25 April 2023, date of current version 9 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270285

RESEARCH ARTICLE

Boosted Gaze Gesture Recognition Using Underlying Head Orientation Sequence

ZAKARIYYA ABDULLAHI BATURE, SUNUSI BALA ABDULLAHI, (Member, IEEE), SUPARAT YEAMKUAN, WERAPON CHIRACHARIT, AND KOSIN CHAMNONGTHAI, (Senior Member, IEEE)

Department of Electronics and Telecommunication Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand

Corresponding author: Kosin Chamnongthai (kosin.cha@kmutt.ac.th)

This work was supported by the Petchra Pra Jom Klao Ph.D. Research Scholarship from the King Mongkut's University of Technology Thonburi, Bangkok, Thailand, under Grant 64/2563.

ABSTRACT People find it challenging to control smart systems with complex gaze gestures due to the vulnerability of eye saccades. Instead, the existing works achieved good recognition accuracy of simple gaze gestures because of sufficient eye gaze points but simple gaze gestures have limited applications compared to complex gaze gestures. Complex gaze gestures need a composition of multiple subunits of eye fixation to contain a sequence of gaze points that are clustered and rotated with an underlying complex head orientation relationship. This paper proposes a new set of eye gaze points and head orientation angles as new sequences to recognize complex gaze gestures. Eye gaze points and head orientation angles powerfully influence gaze gesture formation. The new sequence was obtained by aligning clustered gaze points and head orientation angles with a simple moving average (SMA) to denoise and interpolate the gap between successive eye fixations. The aligned new sequence of complex gaze gestures was utilized to train sequential machine learning (ML) algorithms. To evaluate the performance of the proposed method, we recruited and recorded the eye gaze and head orientation features of ten participants using an eye tracker. The results show that Boosted Hidden Markov Models (HMM) using Random Subspace methods achieved the best accuracies of 94.72% and 98.1% for complex, and simple gestures respectively, which outperformed the conventional methods.

INDEX TERMS Eye-tracking, gaze gestures, head orientation angles, machine learning, sequence recognition.

I. INTRODUCTION

People apply the eyes as one type of input modality for interacting with smart systems such as smartphones, assistive devices, etc. Eyes serve as substitutes for people with hand disabilities and biometric applications [1]. The smart systems capture eye gaze in form of gaze patterns for target selection, gaze prediction, and smartphone control [2]. Gaze patterns are known as gaze gestures and are recognized by machine learning (ML) algorithms like decision trees (DT), template matching (TM), etc. The gaze gestures recognized by the ML algorithms depict the real actions of people [3]. Machine learning algorithms demonstrate good performance to simple gaze gestures like swiping left, straight lines, etc. However, other gaze gestures such as curvy, zigzag, etc. are known as complex gaze gestures and their recognition performance is limited [4]. This is because each complex gaze gesture requires several eye gaze points. These gaze points constitute the beginning and end of complex gaze gestures. Existing works proved that recognition of complex gaze gestures can be precisely achieved using gaze points [5]. However, it is studied that complex gaze

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

gestures have instantaneous angle changes which consist of fewer eye fixations leading to noise in the gaze gestures. The noise is due to uncontrolled rapid eye saccades [6]. Therefore, a suitable ML algorithm is needed to improve the recognition of complex gaze gestures in smart system applications.

In Shi et al. [7], a graph convolution network (GCN) was used to recognize nine simple gaze gestures from gaze points of optical flow approximations. The GCN treated each gaze point of the optical flow as a node and generate edges to connect each successive node. The connected nodes converted the gaze gesture into a graph. GCN convolution layer takes in the graphs and recognizes each through its nodes and self-connected edges. Finally, classification scores were generated by the softmax function to recognize the graphs of nine simple gaze gestures. However, the major limitation of this method is that the generated nodes were clustered at one point of the graph, leading to multicollinearity. The multicollinearity happened due to approximations of most neighboring gaze points in each subunit of eye fixation of the optical flow. These subunits affect the recognition performance of complex gaze gesture. Hence, an effective approach to transform and decompose these subunits is needed. To decompose subunits in complex gaze gestures, we present a new set of gaze points that are obtained from the clustered eye gaze points and head orientation angles, which were absent in [7]. These new sets of gaze point have more distinguishable information and small underlying orientation changes for complex gaze gestures. Each set of gaze points was obtained by selecting a window of ten consecutive gaze points and computing an average gaze point from their respective pixel values. To obtain the next average gaze point, the window selection shift forward to include the next gaze point and leave out the first gaze point. This operation is called simple moving average (SMA) and is performed for all gaze points until the last gaze point. Similarly, SMA process is also performed for the head orientation angles to align the small orientation changes. The new set of gaze points and head orientation angle were combined to form new sequence of complex gaze gesture. The random subspace method was used to randomly select features among the sequence of complex gaze gesture. Boosted HMM was used to learn the sequence of complex gaze gesture from the selected features for recognition. Our contributions to this work are as follows:

i. We measured participants' eye gaze points and head orientation with a single eye tracking sensor from 0.45m to 0.95m having a field of view of 40 × 40 degrees.
ii. We employed simple moving average to reduce redundant fixation density, aligned the sequences of gaze points and head orientation angles, and interpolated the gap between successive fixations.
iii. We adopted a strategy to decompose and restore the fixation from the motion features of the eyes, but the existing method did not consider their influence in causing multicollinearity.

Subsequent sections of this article are as follows: Related works and problem analysis were introduced in section II, and Section III presented the conceptual gaze and head orientation tracking, extraction and transformation of the gaze points and head orientation angles, models of complex gaze gesture, sequence machine learning, models training, and evaluation. Section IV gives the details of the experimental results, performance comparison with baseline method, and ablation studies. Section V provides the discussion. Finally, the conclusion is in section VI.

II. RELATED WORKS

From the existing works, gaze gestures (GG) are widely captured with either one of these three groups of input sensors as depicted in Table 1, depending on the application, availability, and cost. The first group addressed motion-based sensors for GG including EOG which works based on an electric potential difference between the retina and cornea causing changes in the electrostatic field [8]. These position changes were sensed by the electrodes attached to the users' skin closer to the eyes as in [9], [10], [11], and [12]. These systems are wearable, cumbersome, user-unfriendly and the electrodes are biase at some positions. The second group is a multi-modal-based sensors, which combined two or more sensors to capture other input features. The combination of these features can assist in detecting or recognizing some events in the gesture as in [13], [14], [15], and [16]. But these system generally required calibration of the sensors first which make the system to be complex and unfriendly. The third group is Video oculography (VOG) which is the most adopted nowadays because, it can capture the images of the subject eyes, estimate eyes positions, and point of gaze (POG) i.e where the user is looking [17], [18]. The first sub-group of VOG is camera-based, which detects pupil pose and converts them into coordinates [2], [8], [19], [20]. But camera-based suffers from interpolation, head movement, segmentation, identification, and is sometimes cumbersome. The second sub-group is an eye tracker based, the major advantages of using an eye tracker nowadays are: it senses the presence of the subject, locates the positions of the eye(s) precisely, and the POG. Moreover, it is very light, cheap, and comes in two forms depending on the application namely: inversive and non-inversive [6]. The inversive eye trackers are in form of wearable glasses, but wearing these systems [21], [22] is cumbersome, unnatural, and disturbed by motions from the users' heads. The non-inversive eye trackers are placed within their trackable distance of operation from the user and can operate excellently [23], [24], [25]. Eye tracker-based system does not need complex segmentation, it gives the transitions of the eyes movements but has maximum tracking distance of operation to detect the human eyes. Table 2 depicted the summary of the available gaze gesture recognition (GGR) methods.

TABLE 1. GGR according to the input modalities.

System/ Authors' name	Brief Methodology	Highlights	Limitations
MOTION-BASED SENSOR			
Raja et al. [9]	PCA, SVM.	Separate dynamic head movement using head angles and reduced frequency features set from millimeter Radar (mmWave).	mmWave Radar is cumbersome. Need to detect small head movement. Occlusion and vibration.
O'Bard & George [10]	Discrete Wavelet Transformation, SVM + KNN + DT.	Extract mean values of EOG signal to Classify gestures.	EOG controller is not user-friendly. Record only one subject data.
HideMy Gaze with EOG [11]	PCA, KNN + LDA + CT + l-SVM + rbf-SVM.	Used EOG electrodes on the smart glass to measure the optical flow of the eye.	The sensor is more sensitive along the horizontal and vertical axis, leading to lower recognition in diagonal gestures.
Hachaj & Piekarczyk [12]	Median filter, PCA, Bagged DTW + NN and RF.	Head motion Segmentation + classification.	VR helmet is unfriendly to the disabled. Computationally burden.
MULTI-MODAL-BASED SENSORS			
Pettersson & Falkman [13]	CNN.	Used HDM and 2-handheld controllers to Predict the intended direction.	Cumbersome.
Yeamkuan & Chamnongthai [14]	Closet point + POI determination.	LOS' intersection between eye gaze + hand in ROI.	Calibration + Suffers vibration from head and hand + Occlusion of targets by hand.
Yeamkuan et al. [15]	Pointing + VOI and POI determination.	LOS' intersection between eye gaze + hand in VOI.	Cumbersome + Calibration.
Yuan et al. [16]	Error-State Kalman Filter (ESKF) + Euler angle rotations + DNN	Estimate the orientation from signals in IMU and World then, used another camera for yaw drift correction as feedback.	Calibration. Cumbersome.
VOG SENSORS BASED			
Camera-based			
System 14Control [8]	Monochrome camera. Convert PAL signals with A/D converter.	Detect pupil pose and convert pupil poses to coordinate.	Slow due to low memory. Cumbersome.
Rozado et al. [2]	Needleman-Wunsch algorithm	Aligned the sequence of the gesture based on dynamic programming.	The gesture must be at a particular partitioned screen area for matching. Interpolation and Head movement.
Chew & Penver [19]	Open CV's Haar Cascade Classifiers.	Identify users' faces. Locate the eyes and their gaze direction.	Require face identification and segmentation. Tolerate minimal head movement.
Dawood & Hussain [20]	openFace + HMM algorithm.	Extract three rotational angles from webcam videos to help classify head gestures.	Some difficult gesture have the lowest accuracy and few samples of datasets.
Inversive Eye Trackers based			
Chadalavada et al. [21]	Simulated Recall Interview + MPC	Trajectories and intersection points between humans and the robot.	Calibration + Not user-friendly. Misclassification of intention.
Kastrati et al. [22]	CNN + PyrCNN+ EEGnet + Xception + IncTime.	Used eye tracker + 128-channel EEG Geodesic Hydrocel system for Binary gesture classification.	Cumbersome + Not user-friendly.
Non-inversive Eye Trackers based			
Alfaroby E. et al. [23]	LSTM	Raw coordinate of eye tracking data.	Lack of processing tool to address the noise of spontaneous object transition.
Bhattarai & Phothisonothai [24]	EyeMMV + naive Bayes	Separated eye movement with EyeMMV.	Head movement constraints. Low accuracy.
Pichitwong & Chamnongthai [25]	3D POG Detection Algorithm.	Eyes LOS meeting positions based on head shifts	Head movement, Complex + calibration.
Rajanna & Hammond [5]	TM	Resample and moved the data points to the origin.	Low accuracy due to few fixation points in curvy gestures for matching.

A. PROBLEM ANALYSIS

To solve the problem of gaze gesture recognition, authors in [7] introduced GCN and utilized gaze points in 2D velocity as node. However, multicollinearity caused by these velocity nodes is more vulnerable in complex gestures because there are more subunits of eye fixations f and saccades s transitions in complex gestures. Each f is a cluster of many gaze points g, the composition of f and s formed gaze patterns G that is learned in the recognition of given target gaze gesture T. Since, neighborhood frames in f formed the majority gaze points and are comparably closed to each other, their corresponding velocity approximations lead to a cluster of most g at some point as in Figure 1(a). Consequently, velocity approximation changed the entire G of the circle gaze gesture as depicted in Figure 1(b). Hence, velocity approximated features lead to multicollinearity and low-performance recognition of a given T in complex gaze gesture. However, small changes in head orientation angle O features, result in a large variation in g as shown in Figure 1(c). Moreover, the head orientation has more subunit

TABLE 2. Gaze gesture recognition methods.

Algorithms name	Brief Methodology	Highlights	Limitations
Dahmani et al. [26]	TM + LBP +2D-CNN	Face images to estimate the gaze direction from both eyes.	Used perfect images as template.
Koochaki et al. [1]	DBSCAN + HMMs + CNN-LSTM.	Extract spatial and temporal features from gaze points.	Calibration + only kitchen materials + Requires gazing at many objects before guessing the action.
Li, J. [3]	DT + NN + GA	Resampled data points, moved to the origin, and computed Euclidean distance with the template.	Misclassification due to mismatch.
G3 [4]	Eliminate rotation invariance + modified $1 recognizer.	Performed simple and complex gaze gesture with Motion target and 2Dpath stimulus.	Took longer time and less recognition rate for complex gaze gesture than simple.
Shi L. [7]	GCN	Treated each gesture as a graph and fed them to GCN. The nodes are formed from the gaze points and then generate edges of the graph.	The multicollinearity of the gaze points affects the recognition rate.
He H. et al. [27]	SVM + DT + Naive Bayes	Eye gaze pattern + head motion for classification.	Low accuracy in motion features (saccades and smooth pursuit).
Shell J. et al. [28]	Crisp Recognizer + Generic FuzzyTL	Construct coordinate positions and angles between them.	Latency in data caused misclassification. + Few data sample.
Marina-Miranda & Traver [29]	CNN + LSTM.	Used CNN for homograph estimation between frames and LSTM for recognition.	Unsatisfactory accuracy. Misclassification due to motion.

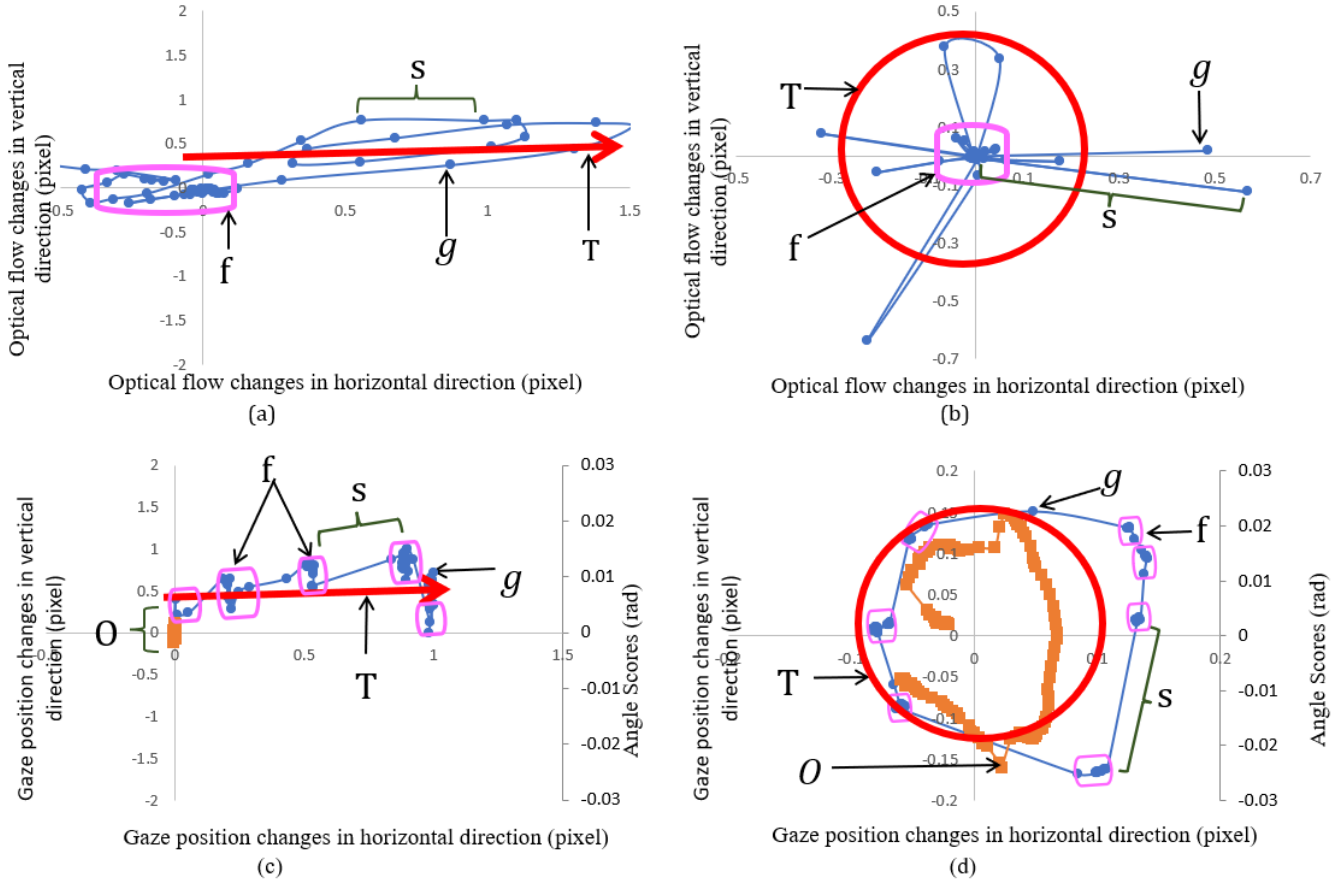

FIGURE 1. Representation of simple gaze gesture (sliding right) and complex gaze gesture (circular), the Red lines describe the expected patterns of the eye gesture in each figure while the respective Blue represent the corresponding gaze points of the eyes: (a) Optical flow to describe sliding right gaze gesture in [7] (b) Optical flow to describe circular gaze gesture (c) Proposed gaze and head orientation to describe sliding right gaze gesture (d) Proposed gaze and head orientation angles to describe circular gaze gesture.

changes and underlying relationship that can explain why complex gaze gesture T are performed well or not as in Figure 1(d).

III. MATERIALS AND METHODS

This section enumerates the proposed method for gaze gesture recognition. It consisted of the following steps:

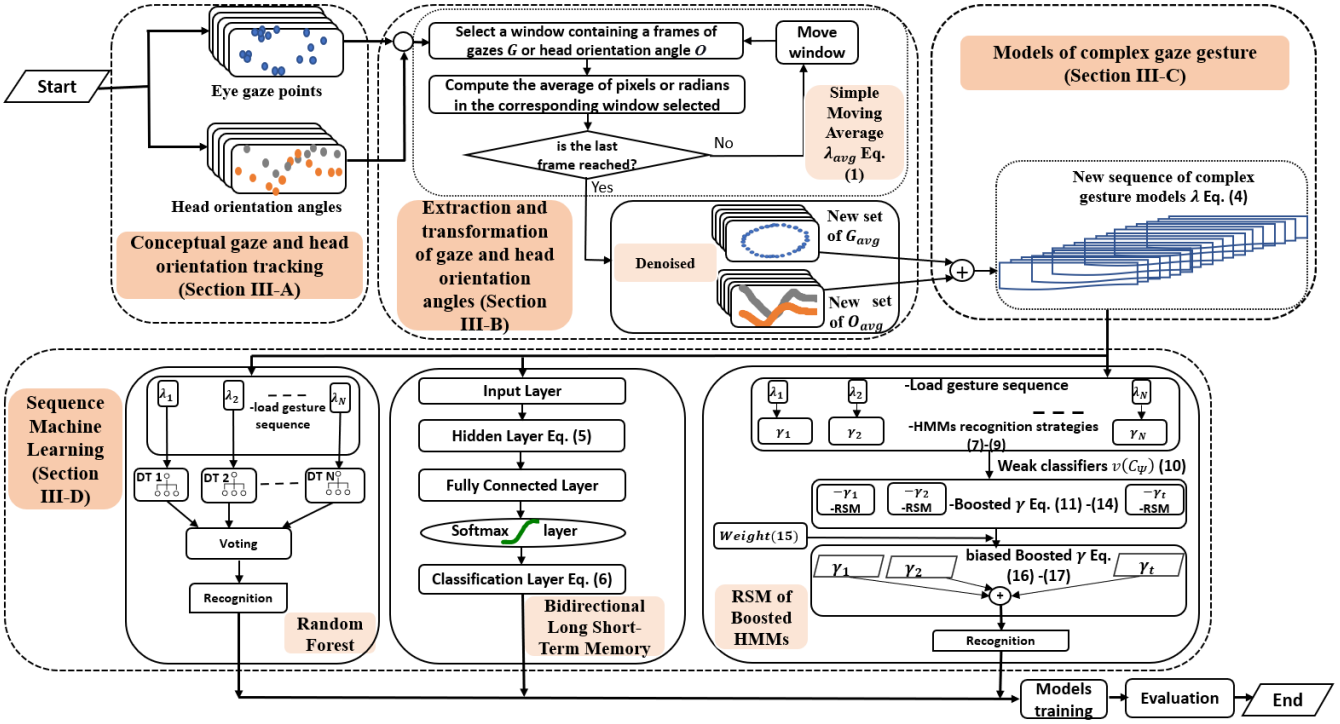

FIGURE 2. Workflow of proposed method.

Conceptual gaze and head orientation tracking, extraction and transformation of gaze points and head orientation angles, models of complex gaze gesture, sequence machine learning, models training, and evaluation. These steps are depicted in Figure 2. The microscopic gaze and head orientation angles of the participants were remotely tracked with a single eye tracking device. SMA was utilized as a processing tool to denoise and aligned the sets of gaze and head orientation angle features. The new sets of features were combined to form a new sequence and utilized to form the models and fed to the sequence ML for complex gesture recognition. Hence, the models training performance and the performance of each ML classifier were evaluated to measure the recognition performance.

A. CONCEPTUAL GAZE AND HEAD ORIENTATION TRACKING

The process of tracking eye gaze and head orientation angle changes is studied by conducting experiments with a single integrated tracking setup. The device tracked the participants eye gaze points and head orientation angles of the participants simultaneously.

1) INTEGRATED TRACKING SETUP

Instead of tracking eye gaze and head orientation separately or using a complex setup, a single eye-tracking sensor was used in this work. Tobii eye tracker 5 was selected to simultaneously track the microscopic changes of both gaze points and head orientation. The eye tracker was placed at the bottom of a computer display and hold firmly using its magnetic flat mount. The Tobii eye tracker 5 has extended features such as 40 × 40 degrees in its field of view, head tracking with six degrees of freedom (6DoF), 133Hz image sampling rate, and continuous gaze recovery. Hence, the participants' eye gaze and head orientation features can be directly measured remotely. TobiiPro.SDK was used to build an algorithm on.Net 4.5 framework of Microsoft visual studio software for data recording on a host computer. Tobii experience software is used to visualize the interaction of the participants on the host computer. The algorithm stored the estimated sequence of gazes and head orientation angle changes in .csv files extension. The complete experimental setup was shown in Figure 3 and their specification was given in table 3.

TABLE 3. Materials specification.

participants	Ten participants
Eye Tracker	Tobii model 5. Trackable distance of operation 45-95cm. Dimension of the tracking box 40×40 degree. Head tracking in 6DoF Image sampling rate of 133Hz
Computer	Intel(R) Core (TM) i5-10300H CPU @ 2.50GHz, RAM 16.00 GB.
software	Tobii experience, TobiiPro.SDK and Microsoft visual studio.
Display	Resolution 1920 × 1020
Target Gestures	Circle, Hover, Infinity, Question Mark, Rectangular, Triangle, X, Z, and Zigzag gestures.

Before the beginning of the experiment, each participant passed through a short calibration eye test with the Tobii eye tracker 5. This is to authenticate each participant and confirm

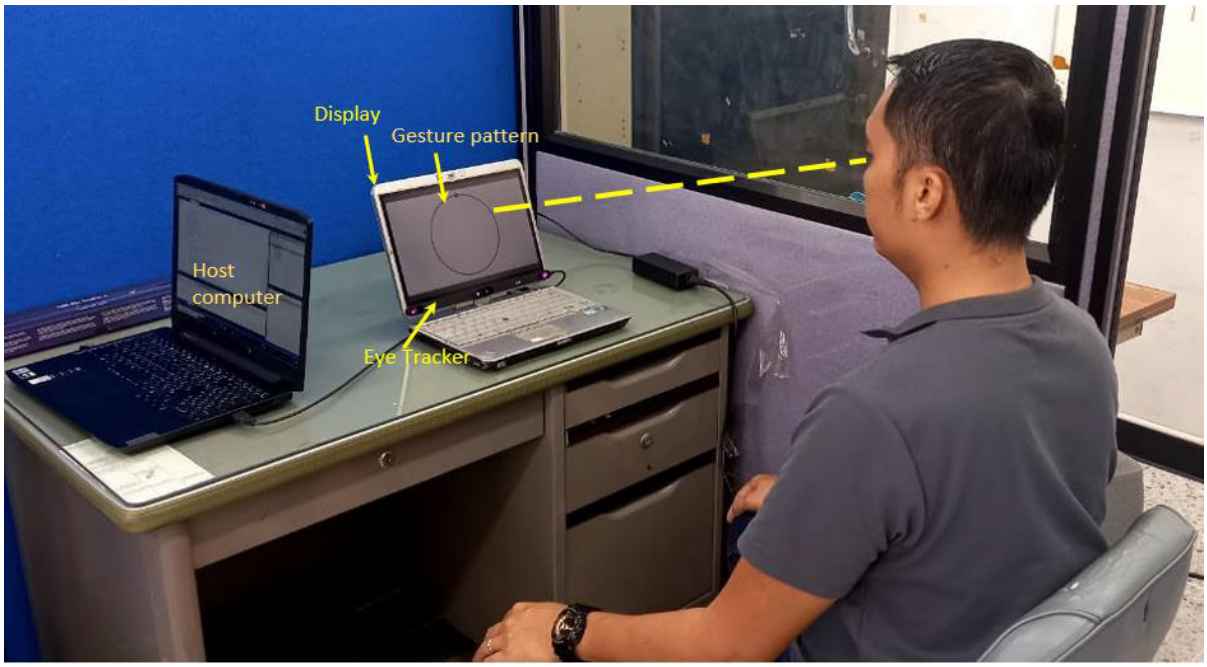

FIGURE 3. Experimental Setup.

the reliability of the data collection. Ten (10) participants between the age of 28-45 years were recruited and were asked to sit on a non-rotary chair. The participants were trained on how to face the eye tracker to have a direct line of sight within a trackable distance of its operation as shown in Figure 3. Two participants have prior experience of interaction with an eye tracker. The participants were allowed to move their heads freely for comfort but must be within the tracking box of the eye tracker. The participants were trained to perform the selected complex gaze gestures by moving their eyes to mimic the designed target gesture T on a display. The estimated input features G and O of each participant were recorded on the host computer in pixels and radians respectively. Each participant performed at least three (3) times for every nine (9) selected classes C of complex gaze gestures. We considered a total sample S of 270 complex gaze gestures. From input features considered, G contained the gaze points in horizontal g_x and vertical g_y directions respectively. Similarly, head orientation angles consists pitch o_p and yaw g_y angles of the head respectively. There is variation in the time taken to complete a particular target T with the experience of each participant. Hence 4-5 seconds was utilized to extract 250 timestamps in each input features.

2) GAZE AND HEAD ORIENTATION SEQUENCE

For every single complex gaze gesture recorded, it contains several eye gaze points and head orientations angle changes of the participants. The gaze point sequence contains the corresponding pair of horizontal $g_x(g_{x_1}, g_{x_2}, \ldots, g_{x_n})$, and vertical $g_y(g_{y_1}, g_{y_2}, \ldots, g_{y_n})$ sequences. Similarly, for the head orientation angles, it contains pair of head pitch angle $o_p(o_{p_1}, o_{p_2}, \ldots, o_{p_n})$, and yaw angle $o_y(o_{y_1}, o_{y_2}, \ldots, o_{y_n})$ sequences. However, some eye gaze points clustered in each subunit of eye fixation and then sparsely separated due to eye saccade or head orientation changes. Small change due to eye saccades or head orientation angles causes a wide variation between successive gaze points and changes their directions. Nevertheless, both sequences of gaze points and head orientation angles play a vital role in the recognition of gaze gestures. Hence a transformation of eye gaze points and head orientation angle features are needed.

B. EXTRACTION AND TRANSFORMATION OF GAZE POINTS AND HEAD ORIENTATION ANGLES

Head orientation plays an important role in linking with an eye to focus and form T correctly, by stabilizing line of sight (LOS) to keep gaze and head shift in balance [30]. Failure to link the two may lead to poor performance of T. Hence, head orientation is vital in stabilizing complex gaze information during T positioning and selection [31]. The work measured gaze points and head orientation angle features using an algorithm built on the .Net framework of Microsoft visual studio. The measured sequence of horizontal gaze points g_x, vertical gaze points g_y, head pitch angle o_p, and head yaw angle $o_y(o_{y_1}, o_{y_2}, \ldots, o_{y_n})$ were saved on a host computer. Due to the rapid jumping of the eye which appears in the extracted gaze sequences $G(g_x, g_y)$, thus transformation using simple moving average (SMA) is utilized. The SMA denoise the sequence of $G(g_x, g_y)$ and create more gaze points that interpolates the gaps between successive f. The transformations of extracted sequences $\lambda_i(g_x, g_y, o_p, o_y)$ is performed in the following sections.

1) DENOISING

For each window w selected, the SMA select $i = 1$ to w, sum the pixel coordinates of the gaze points in the selected i to w, and compute average gaze point. The window moves forward to the next frame and evaluated on $i+1$ to $w+1$, until it reaches the last frame n. Similarly, SMA perform the same moving averaging by taking each frame of head orientation angles in radian. The SMA returned the averaged and aligned sequences of gaze points and head orientation in each frame of the sequence. We selected the window size of 10 frames and Figure 4 depicted each input sequence and it corresponding output of the SMA. The function of denoising the extracted gaze points $G(g_x, g_y)$ and head orientation angle features O is given in eq. (1) [32].

$$\lambda_{avg} = \frac{1}{w}\sum_{i}^{w} \lambda_i \tag{1}$$

where λ_{avg} is the denoised input signal, λ_i is the extracted input signal g_x, g_y, o_p and o_y evaluated on selected window size w of the selected frames on each feature, for $i < w \leqslant n$.

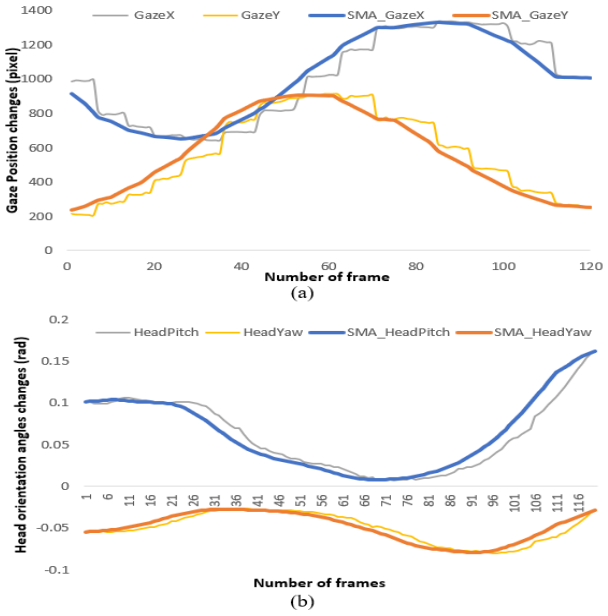

FIGURE 4. Simple moving average of: (a) Gaze points and (b) Head orientation angles.

2) OPTICAL FLOW FEATURE DECOMPOSITION

We further utilized the closed-eyes HideMyGaze camera dataset in [33] to evaluate the performance of the proposed method. The dataset contains clustered gaze points obtained from optical flow approximations of simple gaze gestures. The dataset was collected using Pupil-Lab's eye tracking glass. There are nine gesture classes C of T. Gesture "left" and "right" is performed in a horizontal direction. "Up" and "down" gestures are performed in a vertical direction. "One", "three", "seven", and "nine" gestures are performed in diagonal directions, and "squint" gesture is performed by blinking an eye at the center. The dataset contained a total of 835 gesture samples, each sample has mean optical flow information with 60 frames. However, the optical flow is not constant for the eyes gaze points as described in section II. To overcome the variation, the flow features must be decomposed into new gaze points. The new gaze points supplement the lost eye gazes and then transformed the features as in sections III-B. This is achieved by summing flow changes of a given n successive frames as: $\sum_1^n dg = g_1 + \tau_1$. Where, dg, g_1, τ_1, and n represent mean optical flow, transformed eye gaze points, constant that represent the initial eye gaze points, and the number of frames' window selected respectively. This new gaze points g_1 gives more details of G in forming T than its corresponding approximated feature dg. To obtain subsequent gaze points of g_k, we shifted the window to start from the second frame and end at $n+1$ as $\sum_{2}^{n+1} dg = g_2 + \tau_2, (1 \leq k \leq n)$. The constant term τ_k is ignored because it is initial eye gaze points and uniformly added to all frames, they can only shift the position of each gaze point's frame but do not affect the pattern throughout. In general, we evaluated all corresponding k gaze points contained in each frame of all horizontal dg_x and vertical dg_y directions as adopted in [34]:

$$\sum_1^n dg = g_k + \tau_k \tag{2}$$

The first few frames of g_k appeared to be higher in magnitude because, it represents the integral components of dg, and keep reducing to approach zero as the number of optical flows keeps reducing across the components of dg. These first few and last frames g_k mark a precise decision boundary of the beginning and end of the gaze gesture, unlike its corresponding dg features. The sequence of $O(o_p, o_y)$ features used in this work is the estimated head orientation from the eye tracker which is measured in radians and is found to correspond with computed head orientation angles in equation (3). In addition, gaze orientation angles from the dataset in [33] were computed as given in [30]:

$$O = atan2(g_x, g_y) \tag{3}$$

where O is the computed gaze orientation angle, g_x, and g_y are the corresponding transformed optical flow pair gaze points obtained from eq. (2). These features computed the gaze orientation angle at each frame in radians which lies between $-\pi < O \leqslant \pi$. This computed O corresponds to a similar estimated sequence of head orientation angles from the eye tracker. Thus, the dataset obtained from section-A was processed to learn the sequence of complex gaze gesture. The corresponding dataset in [33] was utilized to evaluate the performance of the method for similar models of simple gaze gestures set using ML algorithms.

C. MODELS OF COMPLEX GAZE GESTURES

The processed sequence of gaze points and head orientation angles were used to form three different models of complex gaze gestures. The models are utilized to train machine

learning algorithms for complex gaze gesture recognition. Model 1 ($\lambda_{\mathbf{g}}$), is formed from the sequence of processed gaze points $G_{avg}(g_{x_{avg}}, g_{y_{avg}})$ as input sequence to the ML algorithms. Model 2 (λ_o), consists of head orientation angles $O_{avg}(o_p, o_y)$. Model 3 is formed by combining the processed gaze points λ_g, and head orientation angle features λ_o, to form $\lambda(G_{avg}, O_{avg})$ as follows:

$$\lambda = concate[g_{x_{avg}}, g_{y_{avg}}, o_{p_{avg}}, o_{y_{avg}}] \tag{4}$$

D. SEQUENCE MACHINE LEARNING

The state-of-the-art ML classifiers is used for the sequential learning of λ are as follows:

1) RANDOM FOREST

Random forest (RF) is the combination of many decisions trees DT. The more randomly assorted trees in the forest, the more robust it becomes and yield more accurate results [31]. In this work, we used bagging scheme by randomly choose L sub-samples, to set up sub-samples S_{tl}, from the training samples S_t. Hence, forming RF involves having equal number of DT. Training S_{tl}, involved randomly choosing attributes for splitting the nodes and choose certain features $\sqrt{\lambda}$ out of the λ. The testing samples was used to test each tree and obtained their respective prediction by performing voting among the result of each tree and then select the most vote as output result. These made it immune from the problem of overfitting [35]. The results of this classifier can be found in Table 5 and Table 8 for the conventional and our datasets.

2) BIDIRECTIONAL LONG SHORT-TERM MEMORY (BI-LSTM)

Long short-term memory (LSTM) is a sequence learning algorithm capable of learning long-term dependency in the sequence. The major limitation of the LSTM is learning the sequence only in forward direction [36]. Thus, bidirectional LSTM correlate sequence at any given timestep in both forward and backward. This is due to forward and backward operations of hidden states in hidden layer [37]. The BiLSTM is constructed from five layers. Input sequence layer, hidden states layer, Fully Connected FC layers, a Softmax layer, and a Classification layer. The specification of this network layers is summarized in Table 4.

TABLE 4. Materials specification.

Network Layers	Model Variables	Selection
Input	Sequence length	Longest
	Minibatch size	30
	Input feature size	4 dimensions
Hidden	Bi-LSTM layer	Longest
	Hidden units	64
	Activation function	Softmax
Output	LSTM model	Many to one
	Number of gestures	9

The input sequence layer of the Bi-LSTM network is fed with the new set of sequences of complex gaze gesture λ as explained in section III-B. Each sequence of the complex gesture has four dimensions composing gaze and head orientation angle features with n length. The hidden layer received λ from the input layer and the hidden states split each λ to learn from each new set of gaze points and head orientation angles. The hidden states memorize the previous and future recurring set of gaze points and head orientation angles association. It memorizes all sequences until all sets in the last λ are bidirectionally learned. The prediction probabilities of each sequence are obtained by assigning weight and adding a bias in the hidden layer as given in equation (5) as follows [37].

$$\mathcal{D} = \mathcal{W}_{h\vec{\lambda}}\vec{h} + \mathcal{W}_{h\overleftarrow{\lambda}}\overleftarrow{h} + b_\lambda \tag{5}$$

where $\mathcal{D}$, $\mathcal{W}$,$\vec{h}$, $\overleftarrow{h}$ and b_λ is the predicted probability, sum of the weight, forward hidden states, backward hidden states, and bias of the Bi-LSTM layer respectively. This output prediction is concatenated in the FC layer. The FC layer concatenated all new set in λ to recognize each gesture sequence by multiplying the sequence with weight matrix. The output of the FC layer is fed to the softmax layer. The activation function in the softmax layer normalized the output of FC layer into a prediction probabilities value between [01]. For training the network, we choose thirty minibatch size, Adam optimizer and trained on a single GPU. The Bi-LSTM network was trained with one iteration per epoch for a maximum epoch of one hundred and twenty. We choose an initial learning rate of 0.001 for the Adam optimizer and the learning rate remains constant. The maximum number of epochs is set to be the maximum number of iterations and stopping criteria. The network training stops as the number of iterations reached the maximum number of epochs. Finally, the classification layer utilized the values from softmax layer and estimate cross-entropy loss for the recognition of complex gestures C_M. The final output of this model is formulated in equation (6) [38].

$$\mathcal{D}_t = P(C_M|\lambda) = \frac{e^{\mathcal{D}_C}}{\sum_{i=0}^{C-1} e^{\mathcal{D}}}, C = 1,2,\ldots,C_M \tag{6}$$

where C, and $\mathcal{D}_C$ are the classes of complex gaze patterns and predicted probability classes C_M when gesture features λ from the previous layers are given respectively. This model network is summarized in Algorithm 1.

3) THE RANDOM SUBSPACE METHOD (RSM) OF BOOSTING HMM

HMM γ_t has the ability to learn stochastic process by splitting the sequence into distinct number of states [39]. For input sequence $\lambda^M = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ as described in section III-B, λ_t is the t^{th} input features which is assumed to be emitted in N hidden states $\psi^N = \{\psi_1, \psi_2, \ldots, \psi_N\}$ where $\psi_i \epsilon \Psi^N$ and $\Psi^N = \Psi_1, \Psi_2, \ldots, \Psi_N$ represent the state collection of the γ_t. The ψ hold a probability density function (pdf) to shows the possibility of emitting certain gesture. Each ψ is interacted with one another by transition coefficients.

Algorithm 1 Bi-LSTM

start
Inputs λ, C_{labels} {sequence of complex gaze, labels}
load λ, *and* C_{labels}
Output *accuracy, and precision*
set n *as length of* λ
get n *of each* λ
sort λ *by* n
split λ *into* $(\lambda_{train}, C_{train}) : (\lambda_{test}, C_{test})$
set *minibatch* $= 30$
set $\lambda_{clasess}$ *as classes of complex gaze*
initialize *no. of input features* $= 4$, $h = 64$, *and* $\lambda_{clasess} = 9$
configure *sequenceinputlayer* $=$ *no. of input features*
configure *bilstmlayer* $=$ *no. of* h
configure $FC = no.\ of\ \lambda_{clasess}$
configure $softmax = yes$
configure *classification layer* $=$ *crossentropyex*
set $max.\ epoch = 120$
select *Adamoptimizer, Gpu, and longest n.*
set *no. of iter* $= max.\ epoch$
for *no. of iter* $= 1$, **do**
    train *with* $(\lambda_{train}, C_{train})$
    repeat until *no. of iter* $= max.epoch$
end for
predict $(\lambda_{test}, C_{test})$ *using eq.* (6)
return evaluate metrics
end

Finally, $\gamma_t = (\pi, A, B)$ and its parameters can be the finite classes of the gesture $C = \{c_1, c_2, \ldots, c_M\}$ in the dataset, having M-gestures and N-states as follows [39].

i. π represent the initial state pdf, $\pi = [\pi_i]_{1xN} = [P(\psi_1 = \Psi_i)]_{1xN} (1 \leqslant i \leqslant N)$, where ψ_1 is the first state in the chain.
ii. A represent the matrix of the state transition, $A = [a_{i,j}]_{NxN} = [P(\psi_{t+1} = \Psi_j | \psi_t = \Psi_i)]_{NxN} (1 \leqslant i, j \leqslant N, 1 \leqslant t \leqslant k)$, where ψ_t and $\psi_{t+1}$are the states at t^{th} and $t + 1^{\text{th}}$frames.
iii. B represent the gesture emission matrix, $B = [b_{i,j}]_{NxM} = [P(\lambda_j \ at\ t | \psi_t = \Psi_j)]_{NxN} (1 \leqslant i \leqslant N, 1 \leqslant j \leqslant M)$. It shows the conditioned probability of the gesture λ_k on the state Ψ_j at t^{th}frame.

For λ_t, our aim is to verify the gesture and obtain the decision by estimating the likelihood between C with the target gesture model $\gamma(T)$ and wrong gesture model $\gamma(W)$. The likelihood can be estimated for a participant when the gestures are conditionally independent of each other as:

$$P(C_\Psi | \gamma_i) = \prod_{t=1}^{k_\psi} P(C_t^\psi | \gamma_i),\ \gamma_i \epsilon \{\gamma(T), \gamma(W)\} \tag{7}$$

Generally, scores of the likelihood $P(C_t^\psi | \gamma_i)$ is calculated through forward-backward process [39]. HMM-based gaze gesture recognition is a binary problem, either closed-set or open-set. In the closed-set case, T is recorded to be known, and both the $\gamma(T)$ and $\gamma(W)$ can be learned in the training phase. For $C_\Psi = \{c_1, c_2, \ldots, c_{k_\Psi}\}$, this kind of recognition is carried out based on log likelihood ratio (LLR):

$$LLR(C_\Psi) = \sum_{t=1}^{k_\Psi} \left[log \frac{P(C_t | \gamma(T))}{P(C_t | \gamma(W))} \right]$$
$$If\ LLR(C_\Psi) \geqslant \mu : accepted, Otherwise : reject. \tag{8}$$

In the open-set case, wrong gestures W, are the unknown, and $\gamma(W)$ may not possibly be determined. For a given λ of test observation from W, recognition determines whether the sample belongs to the T in S or not. The length of the frames l may differ thus, this kind of recognition carried out on normalized log likelihood (NLL):

$$NLL\ (C_\Psi) = \frac{1}{k_\Psi} \sum_{t=1}^{k_\Psi} logP(C_t | \gamma(T))$$
$$If\ NLL(C_\Psi) \geqslant \mu : accepted, Otherwise : reject. \tag{9}$$

As explained in section B, the λ_t composed of many distinguishable uneven units in sequence. Thus, by combining the dominance of SMA method and boosting learning power, the whole of gesture sequence can be recognized via boosted HMMs, by which its discrimination ability on sequence learning is robust than single HMM. Since the problem is treated as binary problem, let the positive and negative value represent the T and W respectively. Based on eq. (8) and (9), the decision from each weak learner in the boosted HMMs can be formulated as:

$$v\ (C_\Psi) = \begin{cases} +1, & if\ LLR(C_\Psi)\ or\ NLL\ (C_\Psi) \geqslant \mu \\ -1, & otherwise. \end{cases} \tag{10}$$

The Random subspace method was utilized in this work because it can train on randomly selected features of λ_t instead of the entire features to reduce the low correlation among the sequence. Baum-Welch algorithms is utilized to determine the models' parameters because of its fewer computation and converge easily [40]. For $\gamma = (\pi, A, B)$ with M-gestures and N-states, the training set having U observation is donated as:

$$C = \{C_1, C_2, \ldots, C_U\} \tag{11}$$

where $C_U = \{c_1^k, c_2^k, \ldots, c_k^k\}$ is the u^{th} sequence having k observation frames with independent observation each. The Baum-Welch algorithms focus on fine-tuning and maximizing the parameters of γ as follows:

$$P(C | \gamma_i) = \prod_{u=1}^{u} P(C_u | \gamma_i) = \prod_{u=1}^{u} Pu \tag{12}$$

For observation C_U, we defined the forward and backward variables are defined as $\alpha_t^u(i) = P(c_1^u, c_2^u, \ldots, c_t^u | \psi_t = \Psi_i, \gamma$ and $\beta_t^u(i) = P(c_{t+1}^u, c_{t+2}^u, \ldots, c_k^u | \psi_t = \Psi_i, \gamma$ respectively. The parameters of γ are approximated as follows:

$$\bar{a}_{i,j} = \frac{\sum_{u=1}^{u} \frac{1}{Pu} \sum_{t=1}^{n_{ku-1}} \alpha_t^u(i) a_{i,j} (C_{t+1}^u) \beta_{t+1}^u(j)}{\sum_{u=1}^{u} \frac{1}{Pu} \sum_{t=1}^{k_u - 1} \alpha_t^u(i) \beta_t^u(j)} \tag{13}$$

$$\bar{b}_j(n) = \frac{\sum_{u=1}^{u} \frac{1}{Pu} \sum_{\substack{t=1 \\ s.t c_n}}^{k_u - 1} \alpha_t^u(i) \beta_t^u(j)}{\sum_{u=1}^{u} \frac{1}{Pu} \sum_{t=1}^{n_{ku-1}} \alpha_t^u(i) \beta_t^u(j)} \tag{14}$$

where c_n is the n^{th} $(1 \leq n \leq M)$. All samples in this scheme are treated equally. The weight obtained in boosting learning scheme is employed in the biased Baum-Welch algorithms [40]. For the T includes T samples, the S_t is equal to $\frac{U(U-1)}{2}$. Let $z^T_{i,j}(1 \leq i < j \leq U)$ be the weight of the pair of training samples S_{ti}, S_{tj}, the normalized weight for target sample $C_u(1 \leq u \leq U)$ is evaluated as:

$$z_u = \frac{\sum_{i=uorj=u} z^T_{i,j}}{2 \cdot \sum_{i,j} z^T_{i,j}} \tag{15}$$

The parameters can be approximated again by assigning the weight to the sample C_u.

$$\bar{a}_{i,j} = \frac{\sum_{u=1}^{u} \frac{z_u}{Pu} \sum_{t=1}^{n_{ku}-1} \alpha^u_t(i) a_{i,j}(C^u_{t+1}) \beta^u_{t+1}(j)}{\sum_{u=1}^{u} \frac{z_u}{Pu} \sum_{t=1}^{k_u-1} \alpha^u_t(i) \beta^u_t(j)} \tag{16}$$

$$\bar{b}_j(n) = \frac{\sum_{u=1}^{u} \frac{z_u}{Pu} \sum_{\substack{t=1 \\ s.tc_n}}^{k_u-1} \alpha^u_t(i) \beta^u_t(j)}{\sum_{u=1}^{u} \frac{z_u}{Pu} \sum_{t=1}^{n_{ku}-1} \alpha^u_t(i) \beta^u_t(j)} \tag{17}$$

From eq. (16) and (17), the approximated parameters can discriminatively model the sequence to the extent that complex gesture samples can be proved. We merged the HMMs with RSM boosting learning scheme as summarized in algorithm 2. The model is trained with the following parameter settings: 30 boosting round, 3 states, and 2 continuous density Gaussian mixtures with diagonal covariance matrix output which delivers the best performance.

Algorithm 2 Boosted-HMM with RSM

1: **start**
2: **set** $\lambda^M, A, B, \gamma(T), \gamma(W)$ {input}
3: **Output** $\bar{a}_{i,j}$ and $\bar{b}_j(n)$ {γ_t parameters}
4: **set** Ψ^N {N hidden states}
5: **initialized** γ_t and its variables π, A, B
6: **estimate** eq. (7) {forward likelihood between $\gamma(T), \gamma(W)$}
7: **estimate** eq. (8) {closed-set case}
8: **estimate** eq. (9) {opened-set case}
9: **modify** eq. (8) and (9) to get eq. (10) {boosted HMMs in weak learners}
10: **donate** training set eq. (11)
11: **adjust** γ_t parameters eq. (12)
12: **define** $\alpha^u_t(i)$ *and* $\beta^u_t(i)$ {forward and backward variables}
13: **estimate** $\bar{a}_{i,j}$ eq. (13) and $\bar{b}_j(n)$ (14) {γ_t parameters}
14: **set** $\frac{U(U-1)}{2}$ {positive training data set}
15: **let** $z^T_{i,j}(1 \leq i < j \leq U)$ {weight of training}
16: **estimate** eq. (15) {normalized weight}
17: **assign** eq. (15) in eq. (13) and eq. (14)
18: **re-estimate** $\bar{a}_{i,j}$ eq. (16) and $\bar{b}_j(n)$ (17) {γ_t parameters}
19: **end**

E. MODELS TRAINING

The sequence of the complex gesture was split into the training and testing ratio of 0.9:01. The weak learners in RF classifiers were trained with a bagging scheme for 100 iterations. The RF learns from the randomly selected features and performs voting for the testing samples to form the results of each DT. For training the Bi-LSTM network, thirty minibatch sizes were chosen, and an initial learning rate of 0.001. The network trained for a maximum epoch of one hundred and twenty with one iteration per epoch. The maximum number of epochs is set to be the maximum number of iterations. Figure 5 shows the training performance of the Bi-LSTM. The Boosted-HMM model is trained with 30 boosting round, 3 states, and 2 continuous density Gaussian mixtures with diagonal covariance matrix output. The boosted HMMs were tested with 5-fold cross-validation.

F. EVALUATION

For the evaluation of the proposed method, the confusion matrix was decomposed to form the numerical result. The correctly classified gaze gesture is known as True positive (P_t). The false negative (N_f) is the target gaze gestures classes that the model predicted as wrong gaze gestures while are target gaze gestures. The false positive (P_f) identify gaze gestures that the model predicts belong to a wrong gaze gesture that does not. True negative (N_t) are gaze gestures that the model correctly identified as wrong gaze gestures. Thus, we adopted from [38] to obtained the true-positive rate, false-positive rate, precision, recall, and *f1*-score for each gaze gesture and an average accuracy.

IV. RESULTS

In this section, we report the quantitative recognition performance results of RF, Bi-LSTM, and boosted-HMM algorithms. These algorithms are trained using the dataset from experiment in Section III as well as the HideMyGaze dataset [33]. From the experimental result, we obtained the following results: Table 5 depicts performance results of RF for each complex gaze gesture. The "circle", "infinity" and "zigzag" gestures has the highest recognition performance in RF classifiers while the "hover' and "question_mark" gestures have the lowest recognition performance. Table 6 depicted the evaluated performance results of Bi-LSTM for each complex gaze gestures. The Bi-LSTM recognition performance rate of the gestures significantly increased as shown in Table 6, except for the "circle" and "infinity" gesture which dropped by 33% and 4.17% respectively compared to the performance of RF classifier in Table 5. However, the average performance result of Bi-LSTM is better than the results of the RF reported. Additionally, the performance results of boosted-HMM for complex gaze gesture were quantitatively reported in Table 7. We obtained the highest recognition rate in gesture "Z" while "circle", "infinity", "triangle" and "zigzag" gestures have lowest and similar recognition rates of 92.5%. Nevertheless, the boosted-HMM averagely outperformed the results of both RF and Bi-LSTM by 1.67% and 14.12% respectively.

Similarly, we further use HideMyGaze datasets in [33] to evaluate our method on simple gaze gesture. Table 8 reported the performance results of RF for simple gaze gesture. We found gesture "one" has the lowest recognition performance, followed by the gesture "down". Furthermore, Tables 9 depicted performance results of Bi-LSTM for each

simple gaze gestures. The recognition performance of gesture ''one'' significantly improved by 25% compared to the result in RF of Table 8 though, it is still among the poorly recognized gestures. The general performance of each gesture learned with Bi-LSTM in Table 9, is better than RF in Table 8 except for gesture ''up'' and ''three'' which were excellently learned with RF. Boosted-HMM performance results of simple gaze gestures were reported in Table 10. Gesture ''three'' and ''up'' have similar and lowest recognition rate except in precision and recall. Moreover, boosted HMM average results in Table 10 are better than the results of both RF and Bi-LSTM. In general, the evaluation results show that an average performance of boosted-HMM outperforms all algorithms, followed by Bi-LSTM and RF.

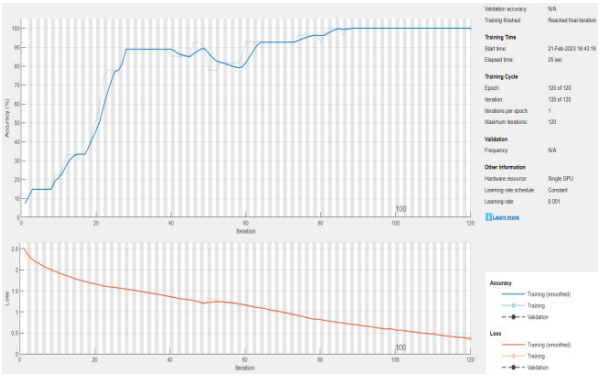

FIGURE 5. Training performance of Bi-LSTM networks on complex gestures.

TABLE 5. Scores per recognition of complex gestures using RF.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
Circle	1	0.02	0.8	1	0.889
Hover	0.5	0.04	0.5	0.5	0.5
infinity	1	0.059	0.5	1	0.667
Question mark	0.625	0	1	0.625	0.769
Rectangle	0.667	0	1	0.667	0.8
Triangle	0.75	0.043	0.75	0.75	0.75
X	0.857	0.043	0.75	0.857	0.8
Z	0.833	0.021	0.833	0.833	0.833
Zigzag	1	0	1	1	1
Weighted Average	0.806	0.022	0.832	0.796	0.798

TABLE 6. Scores per recognition of complex gestures using Bi-LSTM.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
Circle	0.666667	0.333333	0.666667	1	0.8
Hover	1	0	1	1	1
infinity	0.958333	0	1	1	1
Question mark	1	0	1	1	1
Rectangle	0.875	0	1	0.954545	0.976744
Triangle	0.875	0.083333	0.913043	0.913043	0.913043
X	1	0	1	1	1
Z	1	0	1	1	1
Zigzag	1	0	1	0.96	0.979592
Weighted Average	0.930556	0.046296	0.953301	0.980843	0.963264

TABLE 7. Scores per recognition of complex gestures using boosted-HMM.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
Circle	0.925	0.075	0.925	1	0.961039
Hover	0.95	0.05	0.95	1	0.974359
infinity	0.925	0	1	0.948718	0.973684
Question mark	0.95	0.025	0.974359	0.974359	0.974359
Rectangle	0.95	0.025	0.974359	0.974359	0.974359
Triangle	0.925	0.025	0.973684	0.973684	0.973684
X	0.975	0	1	0.975	0.987342
Z	1	0	1	1	1
Zigzag	0.925	0	1	0.925	0.961039
Weighted Average	0.947222	0.022222	0.977489	0.974569	0.975541

TABLE 8. Scores per recognition using RF with transformed motion features.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
One	0.5	0.021	0.9597	0.5	0.657
Three	1	0	1	1	1
Seven	1	0.06	0.94	1	0.969
Nine	0.667	0.021	0.9695	0.667	0.7903
Down	0.641	0.022	0.967	0.641	0.771
Left	0.8	0.075	0.9143	0.8	0.853
Right	0.929	0.028	0.9707	0.929	0.9493
Squint	0.933	0.029	0.97	0.933	0.951
Up	1	0	1	1	1
Weighted Avg.	0.83	0.036	0.858	0.78	0.882

Moreover, the average recognition performance of RF, Bi-LSTM, and Boosted-HMMs of Tables 8-10 was shown in Figure 6. We obtained accuracies of 83%, 87.5% and 98.1% for the RF classifier, BiLSTM, and Boosted-HMMs respectively. The BiLSTM yields an increment of 4.5% accuracy over the RF while Boosted-HMMs achieved 15.1% and 10.6% increment over both RF and Bi-LSTM respectively. The average precision results of RF are 85.8% while the Bi-LSTM achieved precision of 91.67% which is 5.87% less than the precision of the Bi-LSTM. Boosted-HMMs achieved a precision of 100% which is a 14.2% and 8.33% increment over RF and Bi-LSTM respectively. The average recall performance of RF is 78.0% and Bi-LSTM is 95.83% respectively which means RF achieved 17.83% less than the recall of the Bi-LSTM. Boosted-HMMs achieved a recall performance of 98% which is a 20% and 2.17% increment over the recall of RF and Bi-LSTM respectively. The RF, Bi-LSTM, and Boosted-HMMs achieved average F1-scores performance of 88.2%, 92.54%, and 100% respectively. Thus, Bi-LSTM achieved 4.34% F1-scores higher than RF but

TABLE 9. Scores per recognition using Bi-LSTM with transformed motion features.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
One	0.75	0.25	0.75	1	0.857143
Three	0.625	0.375	0.625	1	0.769231
Seven	1	0	1	1	1
Nine	1	0	1	1	1
Down	0.875	0.125	0.875	1	0.933333
Left	1	0	1	1	1
Right	1	0	1	1	1
Squint	1	0	1	1	1
Up	0.625	0	1	0.625	0.769231
Weighted Avg.	0.875	0.083333	0.916667	0.958333	0.925438

TABLE 10. Scores per recognition using Boosted-HMM with transformed motion features.

Gestures	TP Rate	FP Rate	Precision	Recall	F1-Score
One	1	0.001	1	1	1
Three	0.988	0.003	1	0.009	1
Seven	0.988	0.003	1	1	1
Nine	0.997	0.096	1	1	1
Down	1	0	1	1	1
Left	0.946	0.999	1	0.987	1
Right	1	0	1	1	1
Squint	1	0	1	1	1
Up	1	0	1	1	1
Weighted Avg.	0.981	0.008	1	0.98	1

7.46% less than Boosted-HMMs and the Boosted-HMMs achieved 11.8% higher than RF.

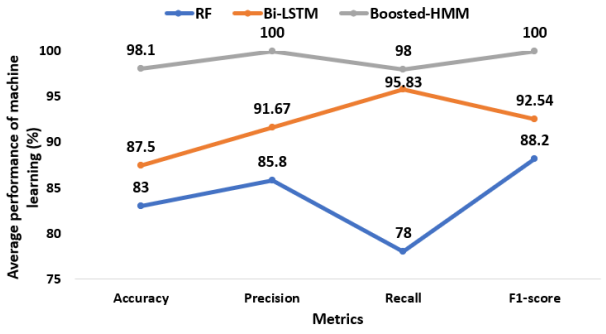

FIGURE 6. Performance comparison results of the ML algorithms of simple gaze gestures using HideMyGaze datasets [33].

In general, the performance of Bi-LSTM and RF is lower than the performance of Boosted HMM. This is due to inherited multicollinearity in the transformed motion feature approximation in the HideMyGaze datasets [33]. However, the combination ability of RSM to automatically select the desired features out of the decomposed motion features and the boosting learning capacity of HMM. Thus, Boosted-HMMs was able to learn all sequences of simple gaze gestures with at least a 94% recognition rate and 98% average performance of all gestures in the HideMyGaze datasets [33].

1) COMPARISON WITH BASELINE METHOD

Table 11 depicts the performance comparison of our adopted method and method Shi et al. [7]. The table 11 also depict the average performance recognition of boosted HMM.

TABLE 11. Performance comparison with baseline method.

Dataset	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Shi *et* al. [7]	RF	88.1	90.77	85.03	Nil
	GCN	97.62	97.18	98.46	Nil
Our approach on Hide My Gaze [33]	RF	83.0	85.8	78.0	88.2
	Bi-LSTM	87.5	91.67	95.83	92.54
	Boosted HMM	98.1	100	98	100
Our Dataset	RF	80.6	83.2	79.6	79.8
	Bi-LSTM	93.06	95.33	98.08	96.33
	Boosted HMM	94.72	97.75	97.46	97.55

The average recognition performance of the RF classifier [7] is 88.1%, 90.77%, and 85.03% for accuracy, precision, and recall respectively. However, the performance results of boosted-HMM with the same dataset are shown in orange color in figure 7. The proposed method achieved

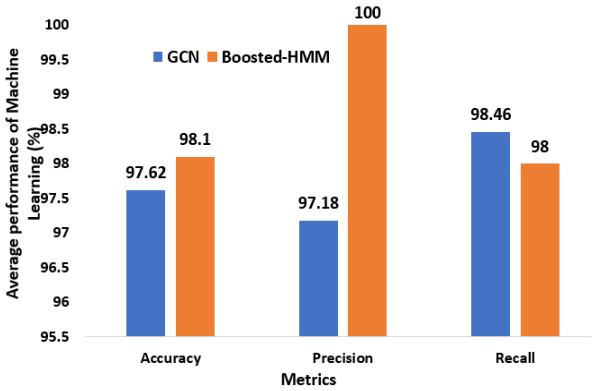

FIGURE 7. Performance comparison between boosted HMMs and GCN in [7].

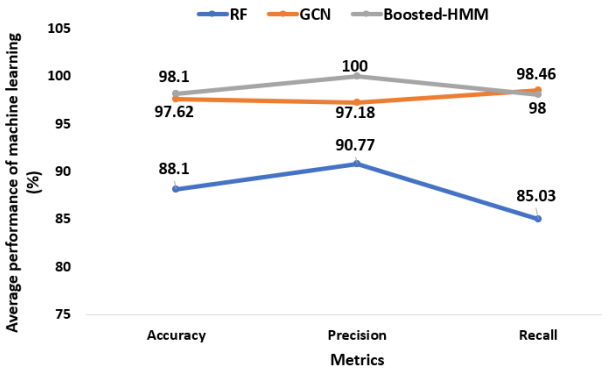

FIGURE 8. Performance comparison between boosted HMMs and ML algorithms in [7].

98.1%, 100%, and 98% for accuracy, precision, and recall respectively. Thus, boosted HMMs outperformed the RF classifier with 10%, 9.23%, and 12.97% in terms of accuracy, precision, and recall respectively. Additionally, the conventional GCN achieved 97.62%, 97.18%, and 98.46% for accuracy, precision, and recall respectively as shown in blue color in figure 7. This means that, boosted HMMs outperformed the GCN with 0.48% and 2.825% in terms of accuracy and precision respectively. Hence, boosted HMMs with 5-fold cross-validation displayed superiority to handle the sequence recognition gaze gestures.

TABLE 12. Performance comparison of three different models of complex gaze gesture with Boosted-HMMs.

Model combination	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Gaze points (Model 1)	59.2	61.41	59.2	59.25
Head orientation angles (Model 2)	29.2	28.1	29.2	28.2
Gaze points + Head orientation angles (Model 3)	94.72	97.75	97.46	97.55

2) ABLATION STUDIES

Table 12 analyzed the performance of three different model combinations with Boosted-HMM. Model 1 consists of gaze points sequence as input sequence to the Boosted-HMMs which achieved 59.2%, 61.41%, 59.2%, and 59.25% for the accuracy, precision, recall, and F1-score respectively. Model 2 consists of head orientation angles as input sequence and achieved 29.2%, 28.1%, 29.2%, and 28.2% for the accuracy, precision, recall, and F1-score respectively. By comparing the performance of model 1 and model 2, the boosted HMMs recognized complex gaze gestures with gaze points better than with head orientation angles. The recognition performance with gaze points is at least 30% better than with head orientation angles in terms of accuracy, precision, recall, and F1 score. Model 3 was formed by combining the gaze points and head orientation angle features. Model 3 achieved 94.72%, 97.75%, 97.46%, and 97.55% for the accuracy, precision, recall, and F1-score respectively. This means that the combinations of gaze point and head orientation angles in model 3 improved the learning of both model 1 and model 2. The Head orientation angles improved the learning of model 1 by 35.52%, 36.34%, 38.26%, and 38.3% in terms of accuracy, precision, recall, and F1-score respectively. Similarly, the gaze points improve the learning of model 2 by 65.52%, 69.65%, 68.26%, and 69.35% in terms of accuracy, precision, recall, and F1-score respectively. Hence, model 3 is the best model which combines the gaze points and the head orientation angles. The performance of these three models were shown in Figure 9. Model 1 is represented with blue color, model 2 is represented with orange color, and model 3 is represented with ash color.

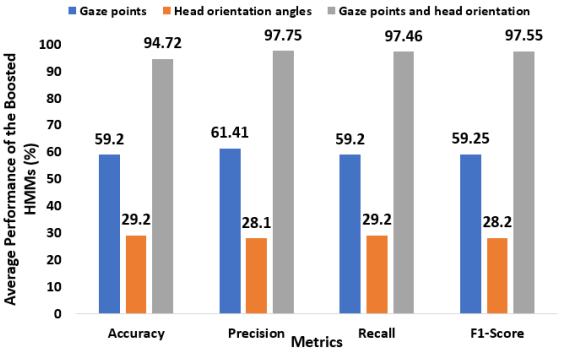

FIGURE 9. Comparison performance of boosted HMMs with three different combinations of complex gesture models.

V. DISCUSSION

The average recognition performance of each classifier for the complex gaze gestures in our dataset is shown in Figure 10. The curves in blue, orange, and ash color represented the average recognition performance of RF, Bi-LSTM and Boosted HMMs respectively. RF classifier and Bi-LSTM achieved recognition accuracy of 80.6% and 93.06% respectively. This means that the BiLSTM yields an increment of 12.46% accuracy over the RF classifier. Boosted-HMMs achieved an accuracy of 94.72% which is a 14.12% and 1.66% increment over RF and Bi-LSTM respectively. The average precision performance of RF is 83.2% while Bi-LSTM achieved a precision of 95.33% thus, RF is 12.13% less than the precision of the Bi-LSTM. Boosted-HMMs achieved a precision of 97.75% which is a 14.55% and 2.42% increment over RF and Bi-LSTM respectively. The average recall result of Bi-LSTM is 98.08% while RF and Boosted HMM achieved a recall of 79.6% and 97.46% respectively. The Bi-LSTM recognition in terms of recall performance is higher than both RF and Boosted HMM by 18.48% and 0.62% respectively. The RF, Bi-LSTM, and Boosted-HMMs achieved average F1-scores performance of 79.8%, 96.33%, and 97.55% respectively. Thus, the precision of Bi-LSTM is 16.53% higher than the precision of RF but 1.22% less than Boosted-HMMs. And the precision of Boosted-HMMs is 17.75% higher than the precision of RF. Hence, Boosted-HMMs is suitable to learn all sequences of complex gaze gestures and achieve a recognition rate of at least 92% for each gesture. The high recognition performance is achieved due to the automatic feature selection ability in each sequence

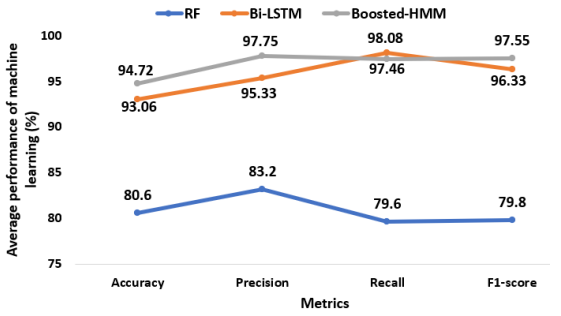

FIGURE 10. Comparison results of the ML performance of complex gaze gestures.

via RSM and the boosting power of the model. However, the RF classifier has the lowest recognition performance of less than 85% in terms of accuracy, precision, recall, and F1-score.

Figure 11 depict gestures with low recognition performance from the results of RF, Bi-LSTM, and the boosted-HMM in Table 5, 6, and 7. The gaze gestures are mostly curvy except ''triangle''. Most of this was due to bad performance by the participants. Circle gaze gesture is badly recognized by Bi-LSTM having a recognition rate of 66.67% and the least recognition rate in Boosted HMMs with 92.5%. The ''circle'' gesture achieves low recognition performance due to the misclassification of the circle with a triangle gesture. we obtained the least recognition performance of 50% in the RF classifier on hover gestures while boosted HMMs achieved up to 95% for its recognition. Infinity gesture was least recognized by the Boosted HMMs with a recognition rate of 92.5% due to some bad samples from the participants. Question mark gesture was poorly recognized by RF with a recognition rate of 62.5% but is well recognized by both LSTM and Boosted HMMs models. The samples of triangle gesture were misclassified with circle gesture and hover, thus the low recognition rate in RF, Bi-LSTM, and Boosted HMMs.

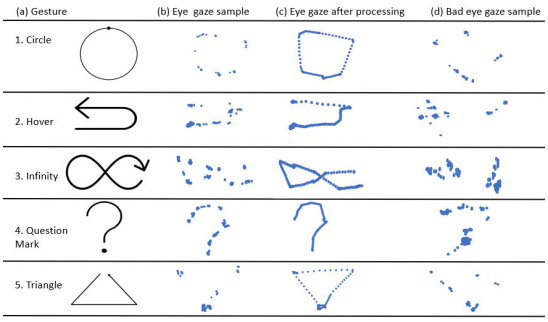

FIGURE 11. Five (5) selected complex gaze gestures: (a) Target gaze gestures (b) Good gaze patterns (c) Processed gaze patterns (d) Bad gaze patterns.

VI. CONCLUSION

In this work, we proposed a new sets of gaze points and head orientation angles to recognize complex gaze gestures. We observed the effect of clustering in the neighborhood frames caused by eye fixations in motion features. An effective decomposition and transformation of SMA approximation were utilized for aligning the clustered sets. We integrate and improve HMMs with RSM boosting learning due to its featuring high discrimination learning ability. Finally, the complex and simple gaze gestures are recognized based on the subunit learning results of the boosted HMMs. Hence, our method may be used in developing gaze interfaces regardless of the types of gaze gestures which extended the applications. However, boosted HMM took long time to boost the learning of weak classifiers. Further research should investigate an effective method of decomposing and restoring the original eye position values from the motion features.

ACKNOWLEDGMENT

The authors are grateful to the anonymous IEEE Access reviewers for their potential reviews and insightful comments.

REFERENCES

[1] F. Koochaki and L. Najafizadeh, ''A data-driven framework for intention prediction via eye movement with applications to assistive systems,'' *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 974–984, 2021, doi: 10.1109/TNSRE.2021.3083815.

[2] D. Rozado, T. Moreno, J. S. Agustin, F. B. Rodriguez, and P. Varona, ''Controlling a smartphone using gaze gestures as the input mechanism,'' *Hum.-Comput. Interact.*, vol. 30, no. 1, pp. 34–63, Jan. 2015, doi: 10.1080/07370024.2013.870385.

[3] J. Li, S. Ray, V. Rajanna, and T. Hammond, ''Evaluating the performance of machine learning algorithms in gaze gesture recognition systems,'' *IEEE Access*, vol. 10, pp. 1020–1035, 2022, doi: 10.1109/ACCESS.2021.3136153.

[4] W. Delamare, T. Han, and P. Irani, ''Designing a gaze gesture guiding system,'' in *Proc. 19th Int. Conf. Hum.-Comput. Interact. with Mobile Devices Services (MobileHCI)*, Sep. 2017, pp. 1–13, doi: 10.1145/3098279.3098561.

[5] V. Rajanna and T. Hammond, ''A gaze gesture-based paradigm for situational impairments, accessibility, and rich interactions,'' in *Proc. ACM Symp. Eye Tracking Res. Appl.*, Jun. 2018, pp. 1–3, doi: 10.1145/3204493.3208344.

[6] S. Yeamkuan and K. Chamnongthai, ''Fixational feature-based gaze pattern recognition using long short-term memory,'' in *Proc. APSIPA Annu. Summit Conf.*, 2020, pp. 1103–1106.

[7] L. Shi, C. Copot, and S. Vanlanduit, ''Gaze gesture recognition by graph convolutional networks,'' *Frontiers Robot. AI*, vol. 8, pp. 1–6, Aug. 2021, doi: 10.3389/frobt.2021.709952.

[8] M. Fejtová, J. Fejt, P. Novák, and O. Štepánková, ''System I4Control: Contactless control PC,'' in *Proc. 10th Int. Conf. Intell. Eng. Syst. (INES)*, 2006, pp. 297–302, doi: 10.1109/ines.2006.1689387.

[9] M. Raja, Z. Vali, S. Palipana, D. G. Michelson, and S. Sigg, ''3D head motion detection using millimeter-wave Doppler radar,'' *IEEE Access*, vol. 8, pp. 32321–32331, 2020, doi: 10.1109/ACCESS.2020.2973957.

[10] B. O'Bard and K. George, ''Classification of eye gestures using machine learning for use in embedded switch controller,'' in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2018, pp. 1–6, doi: 10.1109/I2MTC.2018.8409769.

[11] R. D. Findling, T. Quddus, and S. Sigg, ''Hide my gaze with EOG!: Towards closed-eye gaze gesture passwords that resist observation-attacks with electrooculography in smart glasses,'' in *Proc. 17th Int. Conf. Adv. Mobile Comput. Multimedia*, Dec. 2019, pp. 107–116, doi: 10.1145/3365921.3365922.

[12] T. Hachaj and M. Piekarczyk, ''Evaluation of pattern recognition methods for head gesture-based interface of a virtual reality helmet equipped with a single IMU sensor,'' *Sensors*, vol. 19, no. 24, pp. 1–19, 2019, doi: 10.3390/s19245408.

[13] J. Pettersson and P. Falkman, ''Human movement direction prediction using virtual reality and eye tracking,'' in *Proc. 22nd IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2021, pp. 889–894, doi: 10.1109/ICIT46573.2021.9453581.

[14] S. Yeamkuan and K. Chamnongthai, ''3D point-of-intention determination using a multimodal fusion of hand pointing and eye gaze for a 3D display,'' *Sensors*, vol. 21, no. 4, pp. 1–31, 2021, doi: 10.3390/s21041155.

[15] S. Yeamkuan, K. Chamnongthai, and W. Pichitwong, ''A 3D point-of-intention estimation method using multimodal fusion of hand pointing, eye gaze and depth sensing for collaborative robots,'' *IEEE Sensors J.*, vol. 22, no. 3, pp. 2700–2710, Feb. 2022, doi: 10.1109/JSEN.2021.3133471.

[16] L. Yuan, C. Reardon, G. Warnell, and G. Loianno, ''Human gaze-driven spatial tasking of an autonomous MAV,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1343–1350, Apr. 2019, doi: 10.1109/LRA.2019.2895419.

[17] A. Kar and P. Corcoran, ''A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms,'' *IEEE Access*, vol. 5, pp. 16495–16519, 2017, doi: 10.1109/ACCESS.2017.2735633.

[18] H. R. Chennamma and X. Yuan, ''A survey on eye-gaze tracking techniques,'' *Indian J. Comput. Sci. Eng.*, vol. 4, no. 5, pp. 388–393, 2013.

[19] M. T. Chew and K. Penver, ''Low-cost eye gesture communication system for people with motor disabilities,'' in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, May 2019, doi: pp. 1–5, 10.1109/I2MTC.2019.8826976.

[20] D. A. A. Dawood and B. A. Hussain, ''Machine learning for single and complex 3D head gestures: Classification in human-computer interaction,'' *Webology*, vol. 19, no. 1, pp. 1431–1445, Jan. 2022, doi: 10.14704/web/v19i1/web19095.

[21] R. T. Chadalavada, H. Andreasson, M. Schindler, R. Palm, and A. J. Lilienthal, "Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101830, doi: 10.1016/j.rcim.2019.101830.
[22] A. Kastrati, M. B. Plomecka, R. Wattenhofer, and N. Langer, "Using deep learning to classify saccade direction from brain activity," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, May 2021, pp. 1–6, doi: 10.1145/3448018.3458014.
[23] M. Alfaroby E., S. Wibirama, and I. Ardiyanto, "Accuracy improvement of object selection in gaze gesture application using deep learning," in *Proc. 12th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2020, pp. 307–311, doi: 10.1109/ICITEE49829.2020.9271771.
[24] R. Bhattarai and M. Phothisonothai, "Eye-tracking based visualizations and metrics analysis for individual eye movement patterns," in *Proc. 16th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2019, pp. 381–384, doi: 10.1109/JCSSE.2019.8864156.
[25] W. Pichitwong and K. Chamnongthai, "An eye-tracker-based 3D point-of-gaze estimation method using head movement," *IEEE Access*, vol. 7, pp. 99086–99098, 2019.
[26] M. Dahmani, M. E. H. Chowdhury, A. Khandakar, T. Rahman, K. Al-Jayyousi, A. Hefny, and S. Kiranyaz, "An intelligent and low-cost eye-tracking system for motorized wheelchair control," *Sensors*, vol. 20, no. 14, pp. 1–27, 2020, doi: 10.3390/s20143936.
[27] H. He, Y. She, J. Xiahou, J. Yao, J. Li, Q. Hong, and Y. Ji, "Real-time eye-gaze based interaction for human intention prediction and emotion analysis," in *Proc. ACM Int. Conf. Ser.*, 2018, pp. 185–194, doi: 10.1145/3208159.3208180.
[28] J. Shell, S. Vickers, S. Coupland, and H. Istance, "Towards dynamic accessibility through soft gaze gesture recognition," in *Proc. 12th UK Workshop Comput. Intell. (UKCI)*, 2012, pp. 1–8, doi: 10.1109/UKCI.2012.6335757.
[29] J. Marina-Miranda and V. J. Traver, "Head and eye egocentric gesture recognition for human-robot interaction using eyewear cameras," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7067–7074, Jul. 2022, doi: 10.1109/LRA.2022.3180442.
[30] H. Liu, S. Fang, Z. Zhang, D. Li, K. Lin, and J. Wang, "MFDNet: Collaborative poses perception and matrix Fisher distribution for head pose estimation," *IEEE Trans. Multimedia*, vol. 24, pp. 2449–2460, 2022, doi: 10.1109/TMM.2021.3081873.
[31] B. Li, B. Bai, and C. Han, "Upper body motion recognition based on key frame and random forest regression," *Multimedia Tools Appl.*, vol. 79, nos. 7–8, pp. 5197–5212, Feb. 2020, doi: 10.1007/s11042-018-6357-y.
[32] S. Hansun, "A new approach of moving average method in time series analysis," in *Proc. Conf. New Media Stud. (CoNMedia)*, Nov. 2013, p. 3, doi: 10.1109/conmedia.2013.6708545.
[33] R. D. Findling, L. N. Nguyen, and S. Sigg, "Closed-eye gaze gestures: Detection and recognition of closed-eye movements with cameras in smart glasses," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11506, 2019, pp. 322–334, doi: 10.1007/978-3-030-20521-8_27.
[34] M. Ponti, T. S. Nazare, and J. Kittler, "Optical-flow features empirical mode decomposition for motion anomaly detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 1403–1407.
[35] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *IEEE Access*, vol. 5, pp. 531–536, 2017, doi: 10.1109/CSE-EUC.2017.99.
[36] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.
[37] A. Shrestha, H. Li, J. Le Kernec, and F. Fioranelli, "Continuous human activity classification from FMCW radar with Bi-LSTM networks," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13607–13619, Nov. 2020, doi: 10.1109/JSEN.2020.3006386.
[38] S. B. Abdullahi and K. Chamnongthai, "American sign language words recognition of skeletal videos using processed video driven multi-stacked deep LSTM," *Sensors*, vol. 22, no. 4, pp. 1–28, 2022, doi: 10.3390/s22041406.
[39] S. W. Foo, Y. Lian, and L. Dong, "Recognition of visual speech elements using hidden Markov models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 607–614, May 2004, doi: 10.1007/3-540-36228-2_75.
[40] X. Liu and Y.-M. Cheung, "Learning multi-boosted HMMs for lip-password based speaker verification," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 2, pp. 233–246, Feb. 2014.

ZAKARIYYA ABDULLAHI BATURE received the B.Sc. and M.Sc. degrees in electronics from Bayero University Kano (BUK), Nigeria, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree in electrical and information engineering technology with the King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand. His research interests include artificial intelligence, computer vision, digital image processing, eye tracking, and human gesture recognition.

SUNUSI BALA ABDULLAHI (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics from Bayero University Kano (BUK), Nigeria, in 2014 and 2018, respectively, and the Ph.D. degree in electrical and computer engineering from the King Mongkut's University of Technology Thonburi, Thailand. His current research interests include computer vision, image processing, artificial intelligence, nonlinear optimization and their applications in human motion recognition, pattern recognition, data analysis, and social signal processing.

SUPARAT YEAMKUAN received the B.Sc. degree in telecommunication engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2006, the M.E. degree in automation engineering from the King Mongkut's University of Technology North Bangkok, Bangkok, in 2017, and the Ph.D. degree from the King Mongkut's University of Technology Thonburi, Bangkok, in 2022. His current research interests include computer vision, image processing, human–computer interaction, and machine learning.

WERAPON CHIRACHARIT received the B.Eng. degree in electronics and telecommunication engineering, the M.Eng. degree in electrical engineering, and the Ph.D. degree in electrical and computer engineering from the King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 1999, 2001, and 2007, respectively. He is currently an Assistant Professor with the Department of Electronic and Telecommunication Engineering, Faculty of Engineering, KMUTT. His research interests include digital image processing and computer vision.

KOSIN CHAMNONGTHAI (Senior Member, IEEE) received the B.Eng. degree in applied electronics from The University of Electro-Communications, in 1985, the M.Eng. degree in electrical engineering from the Nippon Institute of Technology, in 1987, and the Ph.D. degree in electrical engineering from Keio University, in 1991. He is currently a Professor with the Department of Electronic and Telecommunication Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi. His research interests include computer vision, image processing, robot vision, signal processing, and pattern recognition. He is a member of IEICE, TESA, ECTI, AIAT, APSIPA, TRS, and EEAAT. He has served as the Chairperson for the IEEE COMSOC Thailand, from 2004 to 2007, and the President for the ECTI Association, from 2018 to 2019. He is the Vice President-Conference of the APSIPA Association, from 2020 to 2023. He has served as an Editor for *ECTI E-Magazine*, from 2011 to 2015, and an Associate Editor for *ECTI Transactions on Electrical Engineering, Electronics, and Communications* (ECTI-EEC), from 2003 to 2010, and *ECTI Transactions on Computer and Information Technology* (ECTI-CIT), from 2011 to 2016.