Received 3 July 2023, accepted 4 August 2023, date of publication 14 August 2023, date of current version 23 August 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3305255

RESEARCH ARTICLE

IDF-Sign: Addressing Inconsistent Depth Features for Dynamic Sign Word Recognition

SUNUSI BALA ABDULLAHI [1,2], (Member, IEEE), AND KOSIN CHAMNONGTHAI [1], (Senior Member, IEEE)
[1]Department of Electronics and Telecommunication Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bang Mod, Thrung Khru, Bangkok 10140, Thailand
[2]Nigeria Police, Louis Edet House Force Headquarters Garki, Abuja 900211, Nigeria
Corresponding author: Kosin Chamnongthai (kosin.cha@kmutt.ac.th)

This research is supported by King Mongkut's University of Technology Thonburi's Postdoctoral Fellowship Under Research Project ID 27180.

ABSTRACT Inconsistent hand and body features pose barriers to sign language recognition and translation leading to unsatisfactory models. Existing recognition models are built up on the spatial-temporal depth S_p features. Finding suitable expert features for the S_p model is challenging especially for dynamic sign words because many inconsistent features exist across hand motions and shapes. In this article, we propose IDF-Sign: an efficient and consistent S_p model from a spatial-temporal multivariate pairwise consistency feature ranking (PairCFR) approach. The temporal features are obtained by computing the 3D position vector of skeletal hand joint coordinates, while the spatial features were obtained by taking every ten spatial coordinates in the 3D video frames and averaging it and doing so until the end of the frames. The PairCFR was used to rank and select the best S_p model features at different feature thresholds. We employed a threshold selection to compute a mid-point value of each ranked feature according to its weight. The receiver operating characteristics (ROC) scheme was employed to identify the relationship between the sensitive parameters and the S_p features, and the obtained values were utilized as modeling inputs. To verify the IDF-Sign, we design a real-life experiment with a leap motion sensor (LMS) consisting of ten signers with a total of ninety dynamic sign words. LMS provides the depth videos, since depth videos are too dense for the S_p model to treat directly, we read the depth videos in comma-separated files in real time. Extensive IDF-Sign evaluations using machine learning on ASL, GSL, DSG, and ASL-similar datasets prove the Optimized Forest achieved an average recognition performance of 95%, 78%, 65.07%, and 95% of the top-1, respectively.

INDEX TERMS Automatic sign language recognition, depth sensor, feature selection, hand gesture, pattern recognition, image processing, 3D video processing.

I. INTRODUCTION

The need for effective sign language and hand gesture models is rising due to the rapid increase in the hard-of-hearing population of over four hundred million (>400) [1] and the downstream activities such as sign language translation, biometric verification system, human action recognition, and touch-less interfaces. Existing sign language models achieved good performance using either single or/and multiple-depth spatial and temporal (S_p) models of raw hand information or/and reconstructing this information to achieve S_p models. S_p models are suitable to recognize sign language because sign language transcribing is a three-dimensional (3D) activity, however, the recognition performance of S_p models is decreasing to dynamic sign words (Sign). For this article, the 3D features are referred to as depth features. A sign is the significant constructing unit of sign language sentences. The decrease in the Sign performance was due to the presence of many inconsistent features in the S_p models. Inconsistent features pose significant barriers to sign language applications

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

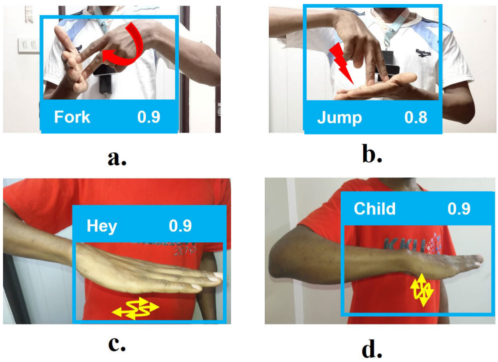

FIGURE 1. Shows dynamic sign word:(a) ASL word *Fork*. (b) ASL word *Jump*. (c) ASL word *Hey* (d) ASL word *Child*. The ASL word in (b) is misclassified with the word in (a), thus as well (c) is misclassified as (d). This happens because of many inconsistent features that provide similar information across the words.

such as misclassification, highly time-consuming network learning, and similarity problems. These problems draw the attention of many researchers for over a decade [2]. Cooper et al. [2] designed depth features using a Microsoft Kinect depth sensor and the features were treated as 3D information using sequential sub-units. Sequential sub-unit models returned an accuracy of 54%. A method to improve this model is known as sequence pattern boosting with a discriminative feature selection strategy [3]. This strategy of feature selection in the S_p sub-units model improved the recognition accuracy to 76%. Still, this method suffers outliers and inconsistent features. Recently, an improved low-cost leap motion sensor is extended to capture 3D skeletal hand joints of Sign [4]. As shown in Fig. 1, the American Sign language (ASL) words [4]; "Fork", "Jump", "Hey", and "Child" are captured by the leap motion sensor as eighty-four 3D skeletal hand joint coordinates. These coordinates are used to compute the hand shape and motion of the sign characteristics. In their method [4], similar Sign words are treated as clusters of 3D sequences, where each cluster contains two words with similar characteristics in shape and motion. Similar characteristics are addressed using the generated temporal dependencies of the 3D fisher vector known as FFV-Bi-LSTM. This 3D Fisher vector sequence contains many inconsistent features which limit the recognition performance of the Bidirectional long-short-term memory (Bi-LSTM) network. Because of the presence of inconsistent features in the FFV-Bi-LSTM model, the model returned a low recognition performance of 91.02% for similar signs, which indicates the significant need for improvement.

Addressing the inconsistent features in the depth models, the feature ranking and selection method demonstrate good performance to better exploits the S_p context [1], [5]. The method achieves promising performance however, easily drops consistent features when fed with complex sign word features as shown in Fig. 1. As shown in Fig. 1, this phenomenon may generate more false positives leading to classification errors. In this article, we extend pairwise consistency feature ranking (PairCFR) capability with the threshold value selection (TV) method to address the problem of inconsistent depth features in the spatial and temporal (S_p) depth model known as IDF-Sign. The temporal features are obtained by computing the 3D position vector of the skeletal hand joint depth coordinates, while the spatial features were computed by taking every ten coordinates in the 3D video frames and averaging it, and doing so until the end of the frames. This is to highlight the gap between the previous and the subsequent frame which is correlated with spatial hand joint indices. The PairCFR was used to rank and select the best S_p model features at different feature thresholds. We employed a threshold value from the threshold selection algorithm to compute the mid-point value of each ranked depth feature according to its weight, in which less significant features will be dropped unbiasedly. The scores provided by the PairCFR-based TV method can be expanded to make selected depth features swing between 0 and 1. This is a significant achievement for the depth features. The benefit of using the PairCFR method is that it yields a score within a narrow range which controls training complexity and uncertainty in modeling.

The receiver operating characteristics (ROC) scheme was computed to identify the relationship between the effective values and the S_p dynamic sign features, and the resulting values were utilized as modeling weights. According to the PairCFR algorithm, the most consistent features in dynamic sign modeling are the hand orientation, velocity, Metacarpophalangeal joints, Proximal joints, and Inter-distal joints. To verify the IDF-Sign, we design a real-life experiment with ten signers with a total of ninety dynamic sign words while suspending the leap motion sensor across their chest to capture the depth videos. We read the depth videos in comma-separated files in real-time using the Laboratory Virtual Instrument Engineering Workbench (LabView) circuit. Extensive IDF-Sign model evaluations using tree-based bagging machine learning classifiers across the public domain data sets demonstrate that the Optimized Forest achieved the best average recognition performance. We itemize the following contributions:

(i) We propose IDF-Sign: efficient and consistent spatial-temporal depth S_p model from a multivariate pairwise consistency feature ranking (PairCFR) using a low-cost depth sensor.
(ii) We improved the PairCFR ranking by integrating the threshold value from the threshold selection algorithm to compute the mid-point value of each ranked depth feature according to its weight, in which less significant features will be dropped unbiasedly.
(iii) The receiver operating characteristics (ROC) scheme was used to determine the relationship between the effective parameters and the S_p dynamic sign features, and the resulting weights were employed as modeling inputs.
(iv) Since depth video frames are too heavy for the model to treat directly, then each frame is automatically

read in comma-separated value files in real-time using programs-subroutines in LabView, without affecting the semantic features of the depth videos.

(v) We treated each depth video frame as a feature instead of using only the segmented features as in the existing methods, this enables us to compute and average every ten coordinates in the 3D video frames and do so until the end of the frames by obtaining a new coordinate.

(vi) We extensively evaluate the proposed IDF-Sign method on challenging 3D GSL, ASL similar and non-similar dynamic sign words as well as 40 phonetically balanced German Sign subsets of HamNoSys phonemes. We achieve state-of-the-art 3D sign word recognition performance.

The sections of this article are as follows: Section II introduces related works, Section III provides problem analysis, inconsistent depth features generation, PairCFR formulation, threshold value selection, and the IDF-Sign recognition is given in Section IV-E. Section IV provides experimental analysis and evaluation. Discussion formed Section V. Conclusions are given in Section VI.

II. RELATED WORK

The challenges of estimating consistent features across video sign language models have been considered for a couple of decades. To identify suitable hand motion features from noisy raw video information, dynamic sequential hand gesture trajectories-based models [6] investigate human hand trajectory feature challenges using sub-unit. Each sub-unit is composed of several correlated consecutive frames. Finally, a dynamic time-warping method was used to obtain a less expensive warping path from the two sub-units. Works [7], [8] proposed a skin blob for hand shape detection, where a threshold was set for skin color. Kurakin et al. [9] introduced sequential hand trajectory recognition using silhouette vector. The hand is tracked from 360 sequences using an action graph, hence graph vertices were considered as the model features. Likewise, the minimum jerk model for dynamic hand movement tracking is adopted in [10] and [11]. However, this model suffers frame-to-frame errors because of hand motion across the frame. Lim et al. [12] proposed a particle filter; the background model is designed from median and mode filters across video frames. Besides, the foreground model is fed into the serial particle filter, thus the models are used to compute the covariance matrix for optimum hand features. While hand trajectories are considered as parameters buried on the Riemannian manifold of shape space used by Smedt et al. [13]. The Riemannian formulation enables consistent hand shape variation estimation. Reference [14] learn trajectories feature vector via Support vector machine (SVM) algorithm. In [15] proposed dynamic sign word recognition of hand trajectory features using the Hidden markov model with Gaussian (HMM-Gauss) method, where each hand is tracked using a region growing scheme. Likewise, Parelli et al. [16] obtain a descriptive motion from the human joint detector by deploying deep convolution pose models. In their work, OpenPose tools provide image pixel coordinates of the hand and body. While Liu et al. [17] propose shifts of hand center and fingertips to represent the hand actions. Recently, many related techniques of hand motion modeling is proposed for sign words such as Camshift [18], [19] to provide a suitable representation of the hand shape and motion, though were limited to shorter segments, as a result, give frame distortion. Convex hull is adopted for effective hand action modeling in [20]. Most of the existing methods contain features that ignore the characteristics relationship of the features. The constraint poses an inevitable challenge for the S_p models to leverage the depth feature context representation. Jesú et al. proposed a Spanish Sign language recognition using Dynamic Time Warping algorithm to implement a 3D LMS-based pattern learning [21]. Besides, researchers are increasingly exploring the use of feature dimensionality reduction in real-life for depth sign language recognition, such as maximal information correlation (MIC) Abdullahi and Chamnongthai [1], and selection of the best subset of generated weights using K-tournament grasshopper optimization (KTGOA) Rim and Kanoun [22]. The former extended the 3D Kalman filter for dynamic hand tracking across the 3D video frames and MIC was used to select the correlated features. While the latter designed a myography force-based ASL signal and used the KTGOA to reduce the size of the Extreme learning machine (ELM) network. However, the former may easily drop consistent features when fed with complex sign word features. This may generate more false positives leading to classification errors. Whereas the latter may ignore the importance of the feature which are the most significant aspect to sign recognition and may lead to a very large network. Das et al. [23] proposed a hybrid model Transfer learning-based convolution neural network (TL-based CNN) with an ensemble tree for the recognition of isolated digits and characters respectively. In the feature vector, a background elimination algorithm is designed that removes the background image features from the sign images. Therefore, a summary of some of the best existing models of sign word recognition is provided in Table 1.

III. MATERIALS AND METHODS

In this section, we describe the strategies for realizing the proposed IDF-Sign method. The proposed method is divided into the generation of depth features for IDF-Sign, the PairCFR method for feature ranking, threshold value selection to improve the PairCFR ranking, and IDF-Sign recognition which describes the metaheuristic and ensemble tree capabilities. The summary of the proposed IDF-Sign is illustrated in Fig. 2.

A. GENERATION OF DEPTH FEATURES FOR IDF-SIGN

In this section, we explain the hand depth information, how it is generated, and its location across the 3D leap motion sensor. This study generated a set of three-dimensional X,

TABLE 1. **Summary of some of the existing sign language models.**

Name and title of the method	Highlight of the method	Performance (%)
Han Sequential hand sub-unit [6]	Hand sub-unit + time-warping	>90
Kurakin et al. [9]	Silhouette vector-based HMM with action graphs	85.2
Lim et al. [12]	Serial particle filter-based covariance matrix	87.33
Smedt et al. [13]	3D Riemannian manifold tracking	68.11 and 82.5
Ghanbari and Seyedarabi [15]	HMM-Gauss with growing scheme	97.48
Dicta-Sign-LSF-v2 [24]	3D upper body, head pose, hand modeling-based CRNN	80.8 and 87.9
Parelli et al. [16]	Descriptive motion using deep convolution pose models	91.38 and 94.56
Liu et al. [17]	3D hand posture evolution tracking with space shift	94.88
Data glove [25]	Accelerometer, gyroscope, and magnetometer-based features with template matching	93.4 words
Roy et al. [19]	Camshift tracking-based skin growing-based trajectories with HMM	77.75
Li et al. [20]	Global enhancement-based perception module with Convex hull tracking	89.82 and 93.10
Abdullahi and Chamnongthai [1]	3D Kalman filter tracking with LSTM	97.98
Jes{ú} et al. [21]	3D LMS-based pattern tracking with DTW	95.17

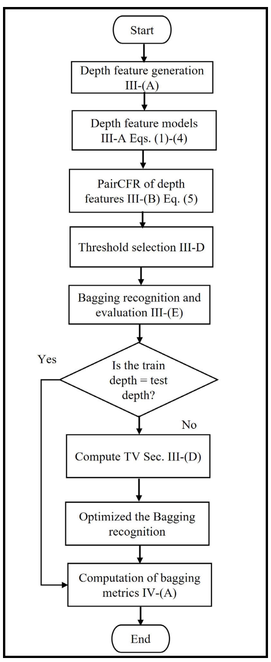

FIGURE 2. **Procedure of the proposed IDF-Sign.**

Y, and Z (where each coordinate is a subset of x, y, and z) features from leap motion sensor that reflect the human daily sign language actions. For the purpose of this study, the 3D features are referred to as depth features. The depth features consist of the following's skeletal hand and finger joint motion and shape coordinates Table 2.

In total, we generate twenty-seven 3D skeletal hand and finger joint coordinates including the tip-to-tip coordinates as presented in Table 2. The first column entitle *feature* describes the features that leap motion sensor extract while the second column entitle *Coordinates* shows the location

TABLE 2. **Hand depth features.**

Features	Coordinates	Features	Coordinates
Hand orientation	Pitch, Yaw, Roll	Proximal joint	Index
Hand velocity	Hand	Distal and inter-distal	Index
Direction	Hand	Metacarpophalangeal	Middle
Direction	Arm	Proximal joint	Middle
Palm to normal	Palm	Distal and inter-distal	Middle
Palm joint	Palm	Metacarpophalangeal	Ringy
Wrist joint	Wrist	Proximal joint	Ringy
Metacarpophalangeal	Thumb	Distal and inter-distal	Ringy
Proximal joint	Thumb	Metacarpophalageal	Pinky
Distal joint	Thumb	Proximal joint	Pinky
Metacarpophalangeal	Index	Distal and inter-distal	Pinky

where each skeletal and finger joints are extracted, which is making a total of eighty-four depth features ((3 × 27) + 3) at different time stamps to achieve the hand temporal features. For an effective sign language model development, we design the temporal and spatial models from the listed 84 features as described in the following section.

1) TEMPORAL DEPTH FEATURE MODEL

In this section, we describe the generation of temporal features and their model development. Section III-A, provides the skeletal hand and finger joints of a total of eighty-four depth coordinate features at different time stamps as a function time to achieve the hand temporal features. These features are vectorially summed up to obtain a temporal model of the hand pattern and motion. The vectors are obtained from the temporal depth skeletal hand joint coordinates by exploiting the 3D position vector operation. The hand shape χ_t is computed using the Eq. (1).

$$\chi_t(x,y,z) = \sum_{\substack{i\in\{x,y,z\}\\ j\in\{x,y,z\}\\ i==j}} [[PYR]_i, [HA]_i, [PW]_i, [M_\wedge]_i, [P_\wedge]_i, \\ \times [IP_\wedge]_i, [D_\wedge]_i, [PI_\wedge]_i, [ID_\wedge]_i + [PN + Dir]_j] \quad (1)$$

where Λ denotes the hand finger capsule which contains the Thumb, Index, Middle, Ringy, and Pinky fingers respectively within a set of *x,y,z*. Where $i == j$ denotes the index of the set which compares the right side with the left side. Where PYR, HA, PW, M_Λ, P_Λ, IP_Λ, D_Λ, PI_Λ, ID_Λ, PN and Dir denotes pitch-yaw-roll, hand-arm direction, palm-wrist, metacarpophalangeal, proximal, inter-phalangeal, Distal, proximal interphalangeal, inter-distal, palm-to-normal, and hand direction joints, respectively. We compute motion features $\vec{M}_t(x_m, y_m, z_m)$ by combining the hand orientation and hand velocity coordinates along the *x,y*, and *z*-axis as described in Algorithm 1. The implementation is described in STEPS 7 to 17 of Algorithm 1. The complete temporal feature vector is a 42-dimension and 84-dimension vector for a single-hand and double-hand, which is obtained as follows.

$$\begin{aligned}\vec{\chi}_t + \vec{M}_t &= (x, y, z) + (x_m, y_m, z_m)\\ &= [x + x_m, y + y_m, z + z_m]\end{aligned} \tag{2}$$

2) SPATIAL DEPTH FEATURE MODEL

In this section, we describe depth spatial information and how it is generated. We generate the spatial and semantic features of the consecutive frames in the temporal model using the weight of frame motion coordinate to exploit the benefits of visual hand features across 3D video frames. The frame coordinate motion weight Eq. (3) defines each video frame as a window W and takes the total sums of the pixel coordinates P_f of the hand and finger joint of the selected window $\sum_{f=1}^{L_w}$. Using the selected window of hand and finger coordinates, we calculate the weighted sum. However, the selected window moves to the next video frame to perform the same operation until it finally reads the last video frame L_w. The weight of the frame coordinate motion yields new features (that is, eighty-four features), these features are built as the spatial model as described in STEP 19 of Algorithm 1. The combination of the temporal and the spatial models gives a comprehensive multiple spatial-temporal depth S_p model (that is, 162 depth features). However, the S_p model consists of one hundred and sixty-eight (168) depth features F in Eq. (4) that are related and non-related sub-feature f to decide the recognition of a sign language word as explained in STEPS 21-22 of Algorithm 1. The non-related features are known as inconsistent depth features and were very insignificant to the recognition performance. Therefore, the insignificant features need to be identified and work accordingly.

$$P_f(x, y, z) = \frac{1}{L_w} . \sum_{f=1}^{L_w} W_f(X, Y, Z) \tag{3}$$

where P_f, L_w, and W denotes the spatial and semantic features of the consecutive frames using the weight of the frame coordinate motion, length of the window, and video frames as windows.

$$F(x, y, z) = \Big\{ \{\vec{\chi} + \vec{M}\}_f + \{P_f\} \Big\} \tag{4}$$

3) GENERATION OF INCONSISTENT DEPTH FEATURES

Inconsistent depth features are defined as the depth coordinates of two or more depth information that contribute to yielding an undesired sign word pattern compared to the originally intended sign word pattern. The major challenges for identifying and correcting inconsistent depth features were not limited to (1) observing the relationships among depth features which may lead to high computation and (2) dropping undesired inconsistent depth features without removing the significant features which may lead to an unconstrained search. For example, the dynamic sign word **FORK** and **JUMP** as shown in Fig. 1 and their feature distribution statistics are illustrated in Fig. 3, which contains depth video frames $V = f_1, \cdots, f_k$ that have similar information I_k and coordinates c. These words are from different classes but exhibit many frames with similar inconsistent depth features, leading to similarity problems. These problems can be expanded in multivariate consistency feature ranking (MCFR) space. MCFR is a heuristics approach needed to constrain the search for inconsistencies to a set of depth features that would provide interesting recognition performance. The MCFR is comprehensively written as a PairCFR function.

B. PairCFR METHOD

From the literature [26], a set of depth features is inconsistent when two or more samples have the same values but different classes. For example, in Fig. 3, the samples of the ASL word **Fork** consist of ThumbMetacarpal, PinkyProximal, PinkyDistal, and RingyProximal joint coordinates belong to class **Fork**, as well as sample values of ThumbMetacarpal, PinkyProximal, PinkyDistal, RingProximal joint coordinates belong to class **Jump** are inconsistent. Therefore, there is a need to drop inconsistent features and harvest the consistent features according to consistency measure. The consistency measure is defined in the multivariate consistency feature ranking (MCFR) [27]. The MCFR discriminates the redundant features using inconsistency measure I. The MCFR is comprehensively written as pair consistency feature ranking (PairCFR) function Γ_k^u when $n \in \{1, \cdots, z\}$ and z denotes depth input features, which can be given as [27]:

$$\Gamma_k^u(n) = \frac{1}{z-1} . \sum_{m \in \{1, \cdots, z\}} \Gamma_k(\{n, m\}) \tag{5}$$

where k denotes a set of depth features having a subset of output class u. Where $\Gamma_k(\{n, m\}) = 1 - I_k(\{n, m\}$ denotes the consistency metric of the subset established by the depth features n and m, which is restricted at $m \neq n$.

Corollary; The inconsistency features $I_k(f)$ have a depth subset feature f within a depth video frame V as given in Eq. (4), where k is obtained from the total sum of all the inconsistency count I_h for all the depth patterns divided by the total number of the depth sequences in k. However, in each given depth pattern (scores of the selected depth features without class), I_h is computed as the total number of the same depth patterns in the depth information minus the number

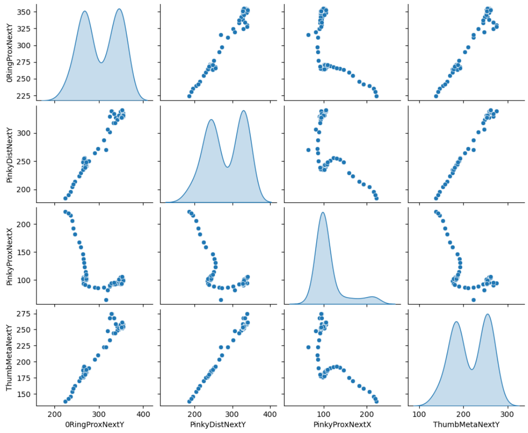

FIGURE 3. Shows distribution of depth features: (**left**; vertical axis) shows ThumbMetacarpal, PinkyProximal, PinkyDistal, RingProximal joint coordinates, and (**bottom**; horizontal axis) show their corresponding joint coordinates from another sign word **Jump**. The coordinates at the top within each frame indicate that the hand shape coordinate is a function of all other parameters.

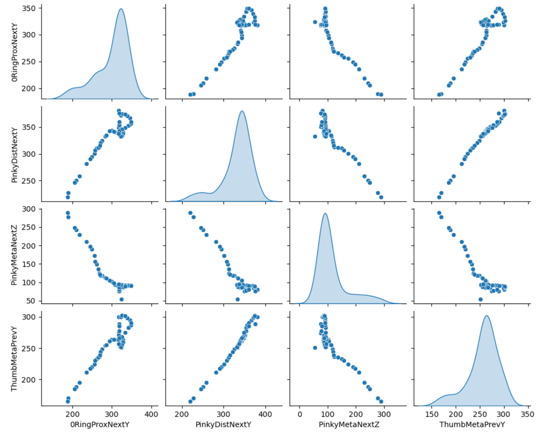

FIGURE 4. Shows distribution of depth features: (**left**; vertical axis) show ThumbMetacarpal, PinkyProximal, PinkyDistal, RingProximal joint coordinates, and (**bottom**; horizontal axis) show their corresponding joint coordinates from another sign word **Fork**. The features concentrated across each frame show that these instances are functions of each other.

of depth sequences of the majority class of the pattern. The final features are analyzed using the threshold value selection (TV), to select the consistently ranked features. To obtain the suitable threshold value of the ranked features, thus a depth feature analysis is performed. The summary of the PairCFR method is shown in Fig. 17.

C. DEPTH FEATURE ANALYSIS

In this article, a feature analysis is conducted to visualize the contribution of each feature within the set. Figs. 3-4 examines the data set by looking at the joint distribution of some pairs of depth features with their instances or sequence length. The instance of the feature suggests that the finger joint feature is a function of all the other parameters. The other features indicate they are functions of each other. Whereas the features concentrated across each frame show that these instances are functions of each other. Thus, the depth features need to be carefully analyzed to avoid dropping significant features.

Therefore, the feature analysis highlights that the ranked PairCFR features are functions of each other to decide the correct classification of the depth skeletal hand joint coordinates. The PairCFR can highlight the solution of recognizing a feature of **Jump** higher than the feature of **Fork** as shown in Figs. 5-7. The blue dots lies in between the top and bottom of the diagonal blue solid line. The closeness of these dots towards the diagonal line shows effective fitness, otherwise not well-fitted. The plot indicates that the model recognizes good with the ROC = 0.5. The plots determine the deviation toward the correct recognition of similar features. Therefore, it provides effective decision thresholds, that may increase the number of true positives. In this regard, we build the model within the tree-bagging classification algorithm. The PairCFR algorithm finds the best ranking and scores of each feature to measure its quality. We improve the ranking fairness of the PairCFR method using the threshold value selection.

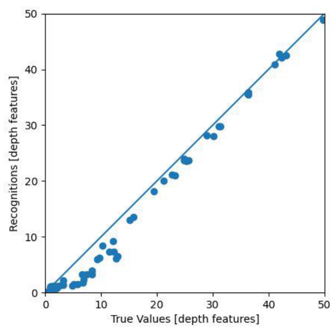

FIGURE 5. Shows fitness of the recognition values versus true values of the depth features: the circle of blue solid shapes lies between the single diagonal blue solid line in the center.

D. THRESHOLD VALUE SELECTION

Threshold value selection is a method of computing the mid-point value of each given depth feature according to its weight using a Threshold value (TV). A TV is computed from the Threshold selector algorithm (TSA). For the purpose of this paper, the TSA algorithm is a heuristics approach that can work by giving a constant TV. The TSA influences the decision of the ensemble tree classification to compute the true positive values equal to or greater than the TV which is set to 0.5. The proposed IDF-Sign method uses the new computed TV as explained in Algorithm 1. We observe the recognition performance of a PairCFR feature matrix with an optimized ensemble forest classification to obtain the suitable TV value. This is done by analyzing the ROC curve to determine the effective probability scores suitable for depth features as shown in Fig. 5. However, the scores that achieve 1 are settled as the final TV to distinguish between the right

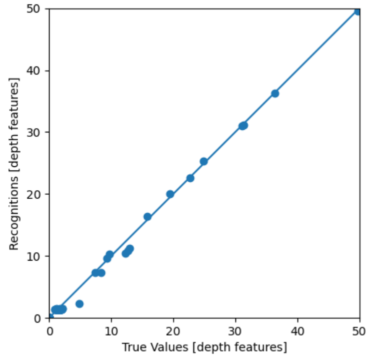

FIGURE 6. Shows fitness of the recognition features versus true depth features: the circle of blue solid shapes lies between the single diagonal blue solid line in the center.

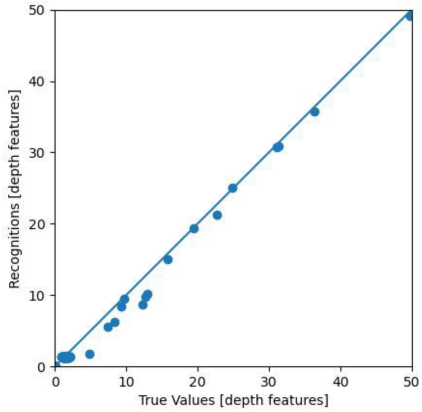

FIGURE 7. Shows fitness of the recognition features versus true features of the depth features: the circle of blue solid shapes lies between the single diagonal blue solid line in the center.

and wrong signs. The ROC measures the rate of true positives against false positives.

E. IDF-SIGN RECOGNITION

In this section, we explain the design of the recognition algorithms in the proposed IDF-Sign. We design tree classifiers from the decision tree-based approaches due to their recognition capabilities. Tree models, such as the decision tree, random forest tree, optimized forest, and rotation forest, are widely applied in sign language recognition [23], [28], [29]. A notable advantage of tree classifiers is that they require the least fine-tuning to settle for the best performance in contrast to Deep Learning classifiers. The main limitation of the tree-based approaches is the restriction while pruning the trees to overcome overfitting. Therefore, large trees fit the noise present in the underlying depth information and hence lead to a low bias and high variance. A decision tree (DT) is the most typical tree classifier which utilized a single decision tree that overfits the training data. Because any slight change in training depth features may lead to a serious change among the resulting decision trees which were generated from the original and modified depth features. To address these trending problems, an ensemble of two or more trees is proposed as a solution in the literature. Thus, we design an ensemble of decision trees to learn logical rules than a single tree. However, when we plant a lot of trees and the final recognition is an average of the output of all the trees in the ensemble, we avoid these problems. However, a large number of trees lead to large memory occupancy and computational overhead [30]. These costs can be crucial for real-life applications such as sign language and hand biometric recognition. It is revealed that not all trees are equally contributed during recognition, meanwhile, some trees may contribute to a downgrade in the average recognition performance of the forest. Therefore, obtaining a subforest (a subset of trees) via the pruning approach may lead to an effective performance than the entire forest [31], [32]. Thus, rotation forests (ROF) can achieve similar or better performance with less number of trees and ensembles. Therefore, we design an optimized Forest (OF) classifier with ROF. The OF classifier is obtained through the strength of the Genetic Algorithm (GA) to obtain an optimal sub-forest from a random forest bagging [31]. The major benefits of bagging with tree methods are (1) Feature scaling has no impact on the structure of the trees (2) The missing values do not affect decision trees (3) The effect of the outliers is very minimal on a decision tree (4) We do not need to do explicit feature transformations to accommodate depth feature interactions.

Bagging generates a new set of depth features $\hat{F}_t$ from the raw depth features F_t. The $\hat{F}_t$ comprised the same number of features and samples as in F_t. However, the samples are generated according to the bootstrap strategy in which more than sixty percent of the samples in F_t were chosen. **Corollary:** a predefined sample c generated during bagging in $\hat{F}_t = \hat{F_{t,1}}, \hat{F_{t,2}}, \cdots, c$ samples, where $\hat{F}_t$ is summarily written as $\hat{F_{t,n}}$ with $n = 1, 2, \cdots, c$ to generate c number of trees for the forest. A random subspace algorithm randomly draws a subset of features f from the whole feature set F to determine the splitting feature for each node of the DT. Let's assume that a forest has R trees, there exist $2^C - 1$ subforests. The most effective way for obtaining the optimal subforests is to train them on the depth training set which improves the generalization performance. Training the subforest by exhaustive search is not realizable as the number of subforests increases substantially. A heuristic is designed to choose a partial search method to effectively search for the best subforests with limited computation capacity [33]. We propose to use a genetic algorithm (GA) that exhibits a high probability to select close-to-optimal subforests, irrespective of the number of actual forests [30], [32]. GA exploits an effective solution through ease-to-use chromosomes Q, which is obtain from the populations W. In the article, the initial population W of GA is fed from high-quality trees R which proved to provide more effective results [30]. Therefore, we intend to compare the recognition performance of the three tree-based bagging classifiers (that is, Random forest, Rotation forest, and Optimized forest) as follows.

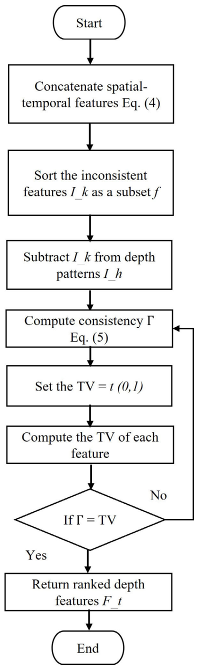

FIGURE 8. **Procedure of PairCFR for IDF-Sign.**

Algorithm 1 PairCFR Feature Learning Algorithm

start
set $\chi(x, y, z)$ {hand pattern}
set i, j, f, L_w {feature indices}
set P_t, $F_t(x, y, z)$ {Target features}
output $\Gamma_k^u(n)$
repeat
  while $i, j = 1 \cdots n$ is detect **for**
  $\longrightarrow$ set $i \in \{x, y, z\}$
  $\longrightarrow$ set $j \in \{x, y, z\}$
  $\longrightarrow$ compare $i == j$
  end while
  compute $\chi(x, y, z)$ in Eq. (1)
  while $\vec{M} \in (x, y, z)$:
  $\longrightarrow$ **if** $\vec{\chi} \in (x, y, z)$:
  $\longrightarrow$ Compute $\vec{\chi} + \vec{M}$ in Eq. (2)
  end while
  compute $P_t(x, y, z)$ in Eq. (3)
  get $F_t(x, y, z) = (\{\vec{\chi} + \vec{M}\} + \{P_{f,t}\})$ from Eq. (4)
  compute $F_t(x, y, z)$ from Eq. (4)
  return $F_t(x, y, z)$
  for $m, n \in F_t$:
  $\longrightarrow$ **if** $m, n \in (1, \cdots, z)$
  $\longrightarrow$ **for** $\forall m \neq n$
  $\longrightarrow$ **compute** $\Gamma_k^u(n)$ in Eq. (5)
  update F_t in Eq. (4)
until F_t is ranked
return $\Gamma_k^u(n)$
end

(i) **Random forest** [31]; An RF classifier comprised a number of tree-structured classifiers $\{\Phi(F, \varrho_t, t = 1, \cdots, n)\}$. Where ϱ_t denotes random vectors that are independently identically distributed. However, a tree can cast a unit vote for the most famous class at sign feature vector F. The RF provides a feature selection during the training of the basic classifier.

(ii) **Optimize Forest** [30]; OF is an ensemble tree-based approach, where a base classifier is optimized from the initial population search of the GA. The structure of the GA is exploited to achieve optimal subforests as explained in Algorithm 2.

(iii) **Rotation forest** ROF [32]; is defined as an ensemble tree scheme, which prevents the defects of the bagging methods to a certain extent. It is assumed that any basic classifier can be developed in ROF.

IV. EXPERIMENTAL ANALYSIS AND EVALUATION

A. EXPERIMENT

The IDF-Sign is experimentally verified to show the performance of the system on real-life sign language recognition. The experiment is conducted with ten signers, who are tasked to perform ninety signs ten times each. The signers are between the age of 24 to 40 years and most of them were right-handed. The signers were requested to hang a leap motion sensor (LMS) at their chest as shown in Fig. 9. The LMS provides the 3D videos of 3D coordinates from complete skeletal hand joints, for details readers are referred to Table 3 and [4]. The LMS is implemented using Microsoft Visual Studio (MS Visual Studio at $\times$ 86 target platform) and.Net framework versions 4.8 and 2.0, which are enabled via the LabView environment as shown in Fig. 10. The complete implementation allows us to address the inefficient data acquisition and processing problems by offering a new 3D video streaming platform with real-time recording computation. The experiment is implemented in three steps. In the first step, We exploit the LeapC which is a C-style application programming interface (API) in the LMS's Software Development Kit (SDK) using the C# binding. LeapC is designed for accessing tracking data from the Hand Tracking Service, as well as creating bindings to C# language as shown in sub-block 1 of Fig. 10. The language bindings are implemented in the native communication (COM) interfaces and the Babel Framework (which is part of Visual Studio SDK). The COM interfaces are managed using the set of wrappers and VSPackages. In the second step, the setup provides the best memory reduction by converting the 3D video frames into a stream of comma-separated files (CSV) in real-time using LabVIEW software as shown in sub-block 2 of Fig. 10. LabVIEW makes system development fast and easy for all. Since 3D videos are too dense for the S_p model to treat directly, thus we consider only the CSV frames in building off the S_p models. We utilized a frame rate of 64 frame-per-second (fps) which is taken along the x, y, and z dimensions. The videos are automatically evaluated using the 3D vector operation as explained in Eq. (2) to capture the hand shapes of the signs. The weight of

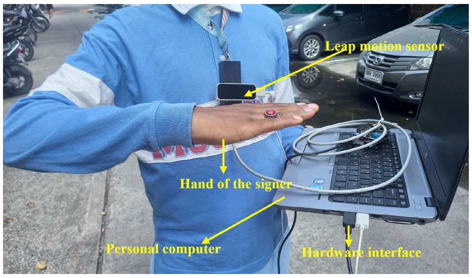

FIGURE 9. Real-life experimental set-up.

the frame motion coordinate function as explained in Eq.(3) was used to generate the spatial frame features. We obtain the complete hand features as given in Eq. (4) to achieve a spatial-temporal hand feature vector. The combined features are scaled using the scaler function to control outliers as given in Eq. (6) and the evaluated frames are readily fed into the PairCFR for feature ranking and selection. In the third step, the selected frames are fed into the PairCFR for the feature ranking and selection. The selected features are built into the bagging classifier for recognition. We further performed experiments with some of the depth baseline data sets to claim the performance of the proposed IDF-Sign across different public domains. Finally, probability scores from the classifiers are evaluated using state-of-the-art evaluation metrics.

B. DATASETS

To claim the effectiveness of the proposed IDF-Sign method, we publicly evaluate this method on the available challenging depth data set as follows.

(i) **Cooper et al.** [3] depth data; The depth data is a Greek Sign Language (GSL) which comprised of 20 dynamic word signs randomly collected from similar and non-similar signs. The data set is performed by six signers, with an average of seven times per sign. Signers were directed to stand in front of an MS Kinect sensor, and all were captured at the same positions.

(ii) **Hanke and Schimaling** [2], [34] depth data; comprised of 40 Deutsche Gebardens-German Sign Language (DGS) performed by fifteen non-native signers for an average of five times each. The depth data is captured with a mobile system with OpenNI to obtain the motion of human skeletal joints during DGS performance. The DGS is the selected subset of balanced HamNoSys phonemes.

(iii) **Abdullahi and Chamnongthai** [4] depth data; comprised of 20 similar dynamic American sign language (ASL) words which are performed by ten signers ten times each. The ASL words are captured using an LMS sensor during motion (That is, the sensor is suspended on the signer's chest).

C. DEPTH FEATURE NORMALIZATION

The selected depth features are further scaled with a function *S(F)* in between the minimum (mn.(F)) and maximum (mx.(F)) feature values to effectively control the effect of some outliers within the depth features F, thus improving the recognition accuracy as given in Eq. (6). In addition, the normalization will prevent the effects of loss of significance, that is the catastrophic cancellation. We complete the final stage of the IDF-Sign method, which is the classifier training.

$$S(F) = \left(\frac{F - mn.(F)}{(mx.(F) - mn.(F))}\right)[mx.(F) \ - mn.(F)] + mn. \tag{6}$$

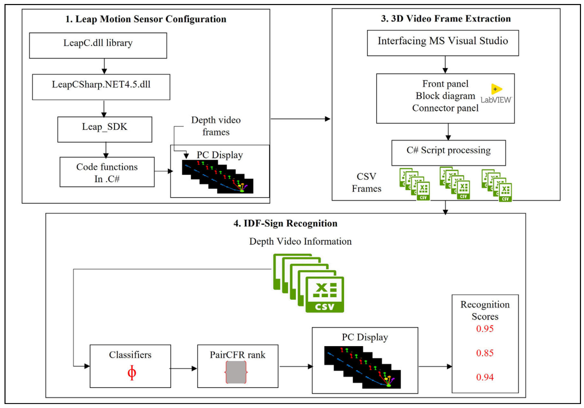

FIGURE 10. Block diagram of experimental implementations.

TABLE 3. Experimental environment.

Systems	Requirements
Personal Computer	Dell G3 CPU: Intel Core i7-9th Gen Memory Size: 8GB DDR4 Solid State Drive: 500 GB
Leap Motion controller	Frame rate: 120 fps Weight: 32g Infrared camera: 2 x 640 x 240 Range: 80 cm FOV: 150 x 120 degrees
Video	64 fps
Signers	10 signers
frequency	10 times per sign

D. CLASSIFIER TRAINING

The classifiers are trained in an end-to-end fashion using the Adam optimizer on a single CORE i7-9th Gen CPU. The classifiers are written in Python 3.10 language with the scikit-sklearn package $http\ :\ //scikit - learn.org/stable/$. We pass the data set as a data frame in the first six steps of Algorithm 2. In lines 8 to 11 we create the random subset of the depth features. The function takes the feature index to compute the number of subsets needed as parameters and outputs f-subsets. With our n_{th} depth features, we set a $f\ -\ value$ to be n, thus we need non-overlapping feature indices and n_{th} iterations. Therefore, the number of iterations (iter) needed is saved in a limit variable in line 13. In such case where the required subset is less than the total number of features, we adopt the first f-entries in the iterable as described in lines 14-19. With the shuffling strategy, we will

TABLE 4. Machine learning settings.

Machine learning requirement	ML design	Selection
Parameters	max. depth	∞
	Batch size	100
	No. of iterations	20
	Population size	20
Technique	search pruning	GA
	No. of Trees	157
	No. of features	Eq. (7)
Ensemble	Bagging	Yes
	Pruning	J48

be returning different volumes at different times. Since we select the subset, thus the available subset in the iterable is deleted to avoid non-overlapping features. Since the subsets are ready, we built our rotation matrix M_R. The rotation matrix is $n \times n$ and develop complete trees to implement the forest performance with the generated depth data frames. We further divide and bootstrap the data frames into training (f_{trn}) 65% and testing (f_{tst}) 35% split to evaluate the performance of the developed models. In line 32 of Algorithm 2, we computed the principal component (PCA) depth features $h_{n,c}$ on the *n-th* subset in f for every c in the *R-th* subset to reduce the large instances of the data frame. We further computed the linear combination of the consistent features to introduce the correlation among the features. The repeated features are considered inconsistent features dropped randomly from both the consistent and redundant features of lines 6 to 10. For the OF, we denote R as the number of trees. Each tree is iterated until we achieved the complete R trees. The achieved h is memorized for each subset. We design an optimized matrix of size $M \times M$, where M denotes the number of depth features. We built the h into the matrix to match the position of the depth features in the raw training set f. In addition, we do the projection $\tilde{f}$ of the f on the optimized matrix via the matrix multiplication. We develop the decision tree with the $\tilde{f}$ depth features. Finally, we memorized the tree as well as the optimized matrix. Therefore, we select 20, 1, 20, and 100, for a number of iterations, random seed, size of the population, and batch size, respectively as illustrated in Table 4. One hundred is chosen as the Size of each bag, as a percentage of the training set size. One hundred is resolved as the Batch size since the base learner is a Batch predictor Y. One hundred is chosen as the number of trees in the ensemble. We have chosen Rotation forest (RF) as the base classifier in the OF. In the OF classifier, we set the number of randomly chosen features using Eq. (7). The OF is designed without representing copies of instances using weights rather than explicitly. The design did not store the out-of-bag recognition. The maximum depth of the trees was set to unlimited. We did not compute attribute importance through mean impurity decrease rather we utilized the row-selected features. The three classifiers are trained with the bagging algorithms. The performance of the classifiers is evaluated using the accuracy and area under the receiver's operating characteristics (AUC ROC). Accuracy fails most of the time to imbalance depth features whereas AUC ROC is a good choice.

Algorithm 2 IDF-Sign Recognition Algorithm

1: **start**
2: **set** R {No. of trees}
3: **set** Q, W {chromosomes and populations}
4: **set** f, n, TV, ROC {training subsets}
5: **set** g, r {crossover and mutation}
6: **set** $h_{n,c}$ {Target features}
7: **repeat**
8: **while** $f = 1 \cdots n$:
9: $\longrightarrow$ set $f_{tst} = 0.35 \times f$
10: $\longrightarrow$ set $f_{trn} = 0.65 \times f$ {TrainFeat}
11: $\longrightarrow$ **set** $h_{g,r}$ {genetic-inspired operator}
12: **end while**
13: **compute** $subset = 0$, $iter = 0$, $limit = \frac{len(iter)}{f}$
14: $\longrightarrow$ **compute** $limit = \frac{len(iterable)}{f}$
15: $\longrightarrow$ **compute** G from Eq. (6)
16: **while** $iter < limit$:
17: $\longrightarrow$ **if** $f \leq len(iter)$:
18: $\longrightarrow \longrightarrow$ Set $subset \longleftarrow f$
19: $\longrightarrow$ **else** $subset = len(iter)$
20: **end while**
21: **repeat** $iter + +$
22: **return** $subsets$
23: **for** $W \in Q$:
24: **adopt** $h_{g,r}$
25: $W_{new} \longleftarrow h_{g,r}$
26: $Q_{new} = 1 \cdots W_{new}$
27: **select best** Q_{new}
28: **return** Q_new
29: **for** $n \in u$:
30: $\longrightarrow$ **compute** $e = dv(f_{trn}, f_{tst})$
31: **for** $subforests \in Q_{new}$:
32: $\longrightarrow$ **compute** $h = pca(subsets)$
33: $\longrightarrow$ **train** $pca.fit(subset)$
34: **train** $model = fit(divide(subsets), subforests)$
35: **return** $model$
36: $\longrightarrow$ **classify** $h = model.classify(f_{tst})$
37: $\longrightarrow$ **select TV** using $TSA.classify(f_{tst})$
38: $\longrightarrow$ **build** $k, h, f = buildOF(\tilde{f}, model, R)$
39: **compute** ROC, TP, AUC
40: **until** $ROC \geq 0.5$
41: **return** $model$
42: **end**

$$RNF = int(\log_2(Y) + 1). \tag{7}$$

where RNF and Y denotes a randomly chosen number of features and a number of available predictors.

E. IDF-SIGN RECOGNITION RESULTS

In this section, we present the recognition results of IDF-Sign. The results of the evaluation metrics are achieved from

TABLE 5. Recognition results of the tree-based classifiers on ASL data set.

Classifiers	No. of depth features	Accuracy (%)	TP (%)	ROC
Random Forest	41	94.22	92.51	0.931
Rotation Forest	41	94.98	93.69	0.937
Optimized Forest	41	95.24	95.109	0.948

the unseen (test) depth features and are presented in average values and the excellent values are emphasized through bold-face. Since we are treating the multi-classification problem, we adopt the One-versus-the-Rest (OvR) multi-class strategy which consists of evaluating a ROC curve per number of sign classes. Always a considered sign is seen as a positive class while the rest a negative class. We use a function [sklearn.preprocess] to binarize the target by one-hot encoding in an OvR fashion. We performed the classification using three different methods as explained in section III. The recognition results of the three classifiers are compared and presented in Table 5. It shows that the OF achieves the best recognition performance on the selected PairCFR-based features of the newly proposed ASL data set. Therefore, we extend the classification of the OF across three different methods. The first method is trained with raw depth features and achieved recognition accuracy of 88.94%, and 89.55% for the complete 84-by-168 single-by-double hand depth features and 84-by-168 single-by-double hand depth features with improved TV-based ROC guide. The IDF-Sign computed an average AUC of 86% as shown in Fig. 11. We analyze the fitness of the modeling parameters as shown in Fig. 12. It shows that the selected features are robust but did not fit well with the chosen modeling classifier. The second method is trained with selected PairCFR depth features and achieved recognition accuracy of 92.77% and 95.24% for the ranked 41 depth features and 41 depth features with an improved TV-based ROC guide as shown in Figs. 13 and 14, respectively. In the ROC evaluations, the IDF-Sign return an average AUC evaluation of 91.02% and 94% for the 41 ranked features and 41 ranked PairCFR features using the TV-based modeling selection. The summary of the IDF-Sign recognition on individual dynamic sign words is illustrated in Table 6. It shows the average computed TPR and ROC of 0.951 and 0.948, respectively. We analyze the fitness of the modeling parameters as shown in Fig. 15. It shows that the selected features are robust and fit well with the chosen modeling classifier. The third method is trained using publicly available depth data sets. Since the best model is achieved from the PairCFR-based feature ranking with OF. The OF achieved the average recognition accuracy of 95.93%, 78%, and 65.07%, for the ASL, GSL, and DSG data sets, respectively as shown in Figs. 16-19.

F. COMPARISON WITH SOME STATE-OF-THE-ART METHODS

We compare the recognition performance of IDF-Sign with some best existing methods on publicly available depth sign language data sets. The evaluation in Eq. (3) is performed across the public data sets to achieve the spatial information. The recognition results of the three data sets are compared with the proposed IDF-Sign. As a result, we achieve the best ensemble tree-based classifier as shown in Tables 7-10. Table 7 presents the individual recognition results of 20 similar dynamic ASL sign words. The results show that the IDF-Sign achieves the best accuracy compared to the FFV-BiLSTM method except at some dynamic sign words; Embarrass, Eight, and Excuse, respectively, where FFV-BiLSTM returns the best accuracy of 100%. Table 8 presents the comparison results between the proposed IDF-Sign and the best existing methods across the DGS data set. The proposed IDF-Sign outperforms the three existing methods [2], [3], [35]. We increase the accuracy up to 2%. The results of comparison between the proposed IDF-Sign and the existing best methods [2], [3], [35] across GSL data sets are provided in Table 9.

The sensitivity of the OF parameters is evaluated using the ROC curve as shown in Figs. 18-19. It is shown that the OF with improved PairCFR manipulations achieves ROC AUC of 0.96, 0.92, and 0.81 for the ASL-similar, GSL, and DGS data sets, respectively.

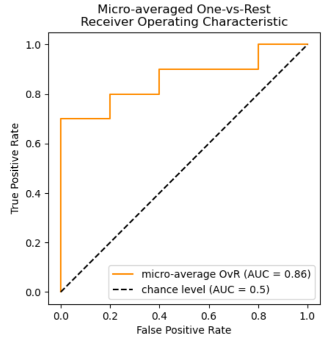

FIGURE 11. Shows the recognition results of 90 ASL words with improved TV modeling using the ROC evaluation curve: the black solid line indicates the ROC margin at 0.5, while the orange solid line indicates the micro-averaging of the single class versus all other classes.

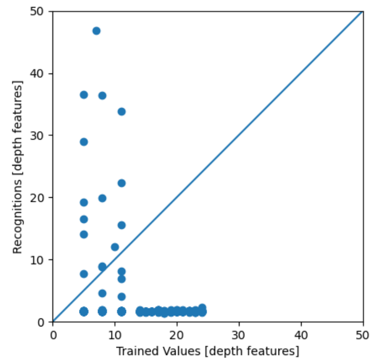

FIGURE 12. Shows the fitness of the classifier parameters across the 84-by-168 depth features with TV-based modeling: the blue solid line indicates the separation between the features, while the blue solid circles indicate the fitness of the data points.

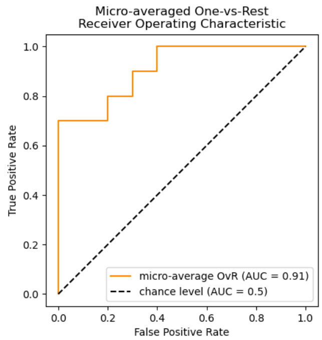

FIGURE 13. Shows the recognition results of 90 ASL words with PairCFR-based modeling using the ROC evaluation curve.

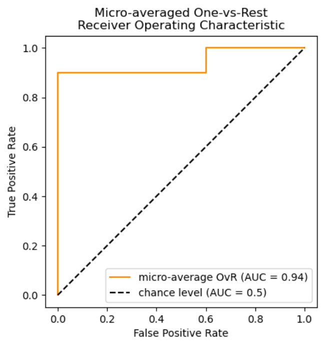

FIGURE 14. Shows the recognition results of 90 ASL words with PairCFR-based TV-based modeling using the ROC evaluation curve.

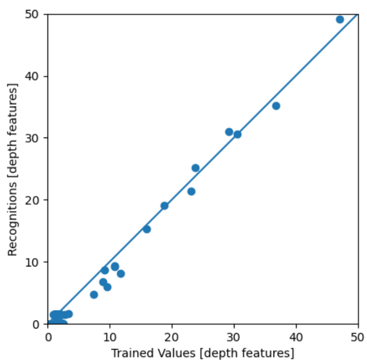

FIGURE 15. Shows the fitness of the classifier parameters across the selected PairCFR depth features with TV-based modeling.

The recognition loss of the trained and tested depth features is illustrated in Figs. 20-22.

Conventional method [3], utilized two strategies for depth feature recognition. The first strategy is the Markov model which uses the feature vector as a whole (that is, 26 features). The second strategy, Sequential Pattern (SP) Boosting performs discriminative feature selection. The SP discriminatively selects the best features from the weak learners. The resulting tree-search method is integrated into a boosting framework; resulting in the SP-Boosting algorithm that combines a set of unique and optimal SPs for a given classification problem. Though the number of selected features

TABLE 6. Recognition performance of IDF-Sign across 90 dynamic ASL words.

TP Rate	ROC Area	Sign words	TP Rate	ROC Area	Sign words
1	1	AGAIN	0.906	0.899	HOT
0.948	0.955	ANGRY	0.883	0.968	HEAR
0.995	0.992	BAD	0.958	0.957	HAIR
0.969	0.881	BICYCLE	0.968	0.948	HOLIDAY
0.974	0.925	BIG	0.916	0.846	INTRODUCE
0.967	0.923	BOX	0.981	1	JUMP
0.974	0.905	BLUE	0.977	0.984	KEY
0.935	0.943	BATH	0.978	0.961	KNOW
0.965	0.968	BROTHER	0.971	0.953	LATIN
0.987	0.977	BETTER	0.938	0.945	LAID OFF
0.985	0.899	BENEFIT	0.815	0.887	LIVE
0.977	0.883	CAR	0.974	0.977	LIKE
1	0.972	CALL	0.903	0.91	MAJOR
0.996	1	CANDY	0.969	1	MANY
0.952	0.967	COMPUTER	0.926	0.989	MEET
0.949	0.95	CARVE	0.913	0.918	MINE
0.963	0.998	CLEAN	0.918	0.885	MOTHER
0.965	0.993	CONVINCE	0.944	0.962	NEED
0.905	0.935	CHECK	0.945	0.945	NIECE
0.964	0.971	COOL	0.927	0.836	NEPHEW
0.985	0.957	CLOTHES	0.993	0.994	ONION
0.916	0.932	COLD	0.982	0.982	OUTLET
0.844	0.947	COME	0.986	0.918	PROFIT
0.882	0.908	DANCE	0.952	0.937	PAST
0.936	0.969	DELICIOUS	0.976	0.974	PAGE
0.933	0.941	DENY	0.966	0.949	PAY
0.986	0.96	DROP	0.977	0.964	PLEASE
0.972	0.972	DRUM	0.984	0.882	READ
0.923	0.981	EAR	0.902	0.915	RED
0.978	0.988	EMBARRASS	0.967	0.994	RETIRED
0.966	0.847	ENDORSE	0.946	0.949	RAILROAD
0.971	0.974	EAGLE	0.978	0.911	RIGHT
0.949	0.967	EGYPT	0.964	0.972	RESPECT
0.911	0.918	FINISH	0.982	0.932	SATURDAYS
0.937	0.965	FATHER	0.949	0.965	SHAVE
0.929	1	FOX	0.975	0.928	SHOES
0.912	0.974	FRUIT	0.993	0.885	SMALL
0.944	0.994	FUNNY	0.957	0.945	SORRY
0.924	0.932	FIRE	0.942	0.958	SUN
0.938	0.916	FEAR	0.935	0.979	SCIENCE
0.913	0.903	FEW	0.955	0.974	STREET
0.955	0.988	GO	0.968	0.968	SISTER
0.908	0.929	GRANDFATHER	0.944	0.957	SURPRISE
0.935	0.95	GRANDMOTHER	0.927	0.928	SHOCK
0.982	1	GOOD	0.949	0.956	SWEAT

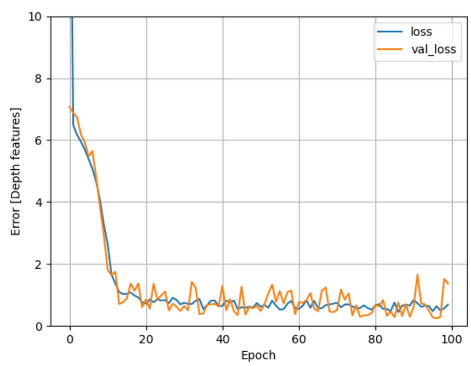

FIGURE 16. Shows the first recognition performance of selected depth features: the blue solid line indicates the recognition loss at 100 epochs, while the orange solid line indicates the validation loss.

of SP boosting is not clearly known by the readers. For this work, classifiers are built in an OVR manner and the results are aggregated for each sign class. The process in Eq. (3) is performed across the GSL data set to obtain other spatial information. However, our proposed IDF-Sign selected the best 14 features which return an average recognition performance of 78% at top-1. It shows that the IDF-Sign is better than the three existing methods [2], [3], [35]. However, the

TABLE 7. Recognition Results of 20 similar dynamic ASL words [4] using IDF-Sign.

Class	Accuracy (%) FFV-BiLSTM [1]	Accuracy (%) IDF-Sign	Class	Accuracy (%) FFV-BiLSTM [1]	Accuracy (%) IDF-Sign
Again	90	95.3	Fork	90	95.9
Angry	90	90.5	Good	90	93.9
Bad	100	99.4	Hey	90	95.2
Child	90	88.2	Happy	80	92.7
Cheap	80	94.4	Hot	90	95.5
Dance	90	99.2	Jump	80	90.7
Eight	100	99.8	Like	90	94.8
Embarrass	100	98.1	Money	90	93.6
Expensive	90	94.9	Read	100	100
Excuse	100	98.7	Short	90	95.3

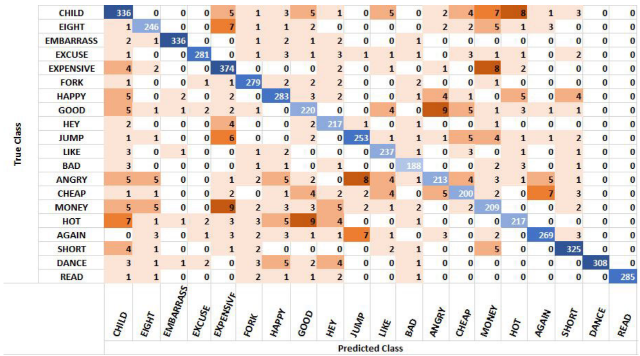

FIGURE 17. Confusion matrix of similar ASL datasets.

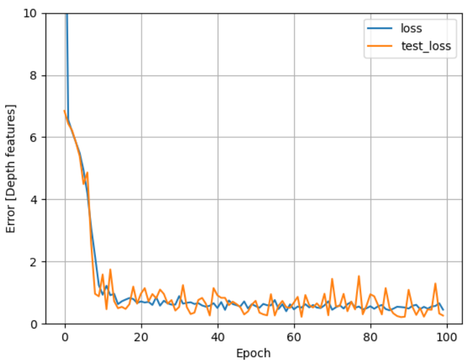

FIGURE 18. Shows the recognition performance of selected depth features of the GSL data set: the blue solid line indicates the recognition loss at 100 epochs, while the orange solid line indicates the testing loss.

TABLE 8. Results of comparison between IDF-Sign with SOT methods on DGS data set at top-1.

Approach	Accuracy (%)
SP + HMM [2]	55.4
Boosted-HMM [3]	49.4
Tr-msHMM [35]	63.1
Proposed IDF-Sign	**65.07**

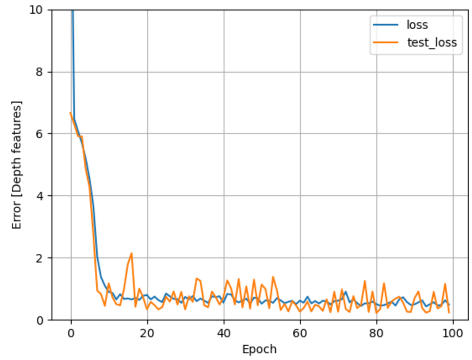

FIGURE 19. Shows the recognition performance of selected depth features of the DGS data set: the blue solid line indicates the recognition loss at 100 epochs, while the orange solid line indicates the testing loss.

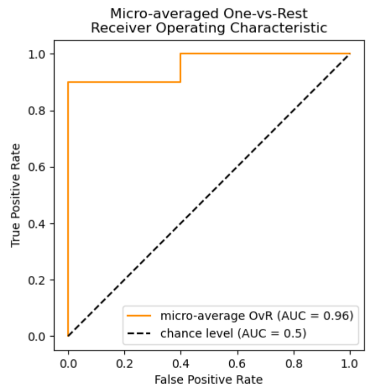

FIGURE 20. Shows the ROC evaluation curve of the ASL data set: the black solid line indicates the ROC margin at 0.5, while the orange solid line indicates the micro-averaging of the single class versus all other classes.

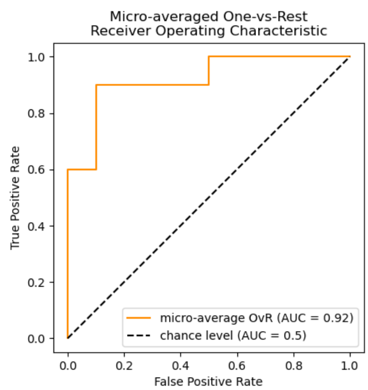

FIGURE 21. Shows the evaluation results of the ROC curve using the one versus rest strategy on the GSL data set: the black solid line indicates the ROC margin at 0.5, while the orange solid line indicates the micro-averaging of the single class versus all other classes.

analysis of the two employed strategies to show the sensitivity of the proposed method is provided in Table 10.

G. ABLATION STUDIES

The resulting ROC curves of the OF are analyzed according to the similar feature count, that is TSA substituted the scores with 0.5. We repeated this method to all the considered sign scores. Therefore, a distinct classifier is realized that recognize sign from the same family according to the mid-point score of 0.5 to distinguish the TP rate from the false positive rate (FP) rate as shown in Fig. 14. The improved OF

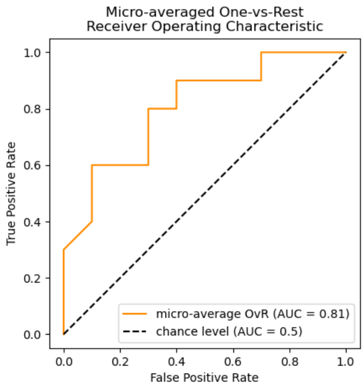

FIGURE 22. Shows the first ROC evaluation curve of the DGS data set: the black solid line indicates the ROC margin at 0.5, while the orange solid line indicates the micro-averaging of the single class versus all other classes.

TABLE 9. Results of comparison between IDF-Sign with SOT methods on GSL data set at top-1.

Approach	Accuracy (%)
SP-boosting [3]	76
Boosted-HMM [3]	54
Tr-msHMM [35]	63.1
Proposed IDF-Sign	**78**

TABLE 10. Results of comparison between IDF-Sign with SOT methods on GSL-20 data set with two strategies.

Strategies		Markov models		SP-boosting		Proposed IDF-Sign	
		Top-1 (%)	Top-4 (%)	Top-1 (%)	Top-4 (%)	Top-1 (%)	Top-4 (%)
SI	1	56	80	72	91	73	94
	2	61	79	80	98	79.9	99
	3	30	45	67	89	71	93
	4	55	86	77	95	78.8	98
	5	58	75	78	98	80.6	98
	6	63	83	80	98	82	99
Avg. Accuracy		**54**	**75**	**76**	**95**	**78**	**97**
SD		**79**	**92**	**92**	**99.9**	**95.3**	**98.99**

algorithm is achieved using bagging with an ensemble forest scheme. The improved OF provides a number of soft decision boundaries which makes the number of TP rates higher in the depth feature recognition as shown in Table 6. This strategy is employed for the public depth data sets which improved the recognition accuracy of 2%, 2%, and 5% for the GSL, DSG, and ASL data sets, respectively. The resulting ROC curve is evaluated using the similar feature frequency, that is for each value in the curve, the equivalent distinct probability score is utilized to count the frequency of similar features present with score $\geq$ TV. The frequency is the reflection of the number of wrongly classified classes. We employed the ROC evaluations to determine the bias from class imbalances and the influence of the FP rates. The developed depth feature models are trained in three different methods as explained in Table 11. The first method employed the 168 raw depth features in the ensemble tree model using OF, which achieve a recognition accuracy of 89.55% with an inference time of 850 ms. The second method adopted 41 ranked depth features from the PairCFR using the OF, which return a recognition accuracy of 92.77% with an inference time of 65 ms. The third method is built up using the selected 41 features of PairCFR-ranked base-TV selection and 95.21% recognition accuracy is achieved with OF. It is demonstrated that the third method returns the best performance when compared to the second and first methods, respectively.

We perform the signer dependence (SD) and signer independence (SI) evaluation strategies to evaluate the effectiveness of the proposed IDF-Sign. For fairness, we adopt the same method of using Top-1 and Top-4 strategies as in [3].

1) SD

To verify the robustness of the proposed IDF-Sign in the SD setting, we randomly divided the datasets into the 65% sample frames for the training, and the 35% are referenced for testing, respectively. Table 10 shows the performance of the IDF-Sign across the six signers. The column at the bottom contains the average recognition results and the IDF-Sign achieves an accuracy of 95.3% and 98.99% at the Top-1 and Top-4 respectively, which is higher than the results of the Markov model and SP-boosting. The proposed IDF-Sign increases the recognition accuracy to 16% and 3.3% at TOP-1 compared to the existing Markov and SP-boosting, respectively. The IDF-Sign increases the recognition accuracy to 6% and conforms with the existing results compared to the 92% of Markov and 99.9% of SP-boosting. It is evident from Table 10 that IDF-Sign achieves a higher accuracy rate compared to existing methods.

2) SI

In a signer-independent strategy, a leave-one-out scheme is utilized to verify the accurate recognition performance of IDF-Sign. This strategy investigates the capability of the classifier in recognizing new depth sign features. The recognition accuracy analysis of the proposed IDF-Sign and state-of-the-art methods are presented in Tables 6, 7, and 10 with the SI setting. Specifically, IDF-Sign attains an increase in accuracy rate from 54% to 78% on Top-1, 75% to 97% on Top-4, 76% to 78% on Top-1, 95% to 97% on Top-4, over GSL data set on Markov and SP-boosting respectively. Furthermore, IDF-Sign yields a 24% and 2% increase as compared to Markov and SP-boosting on the GSL dataset. The results analysis shows that the raw depth model and PairCFR model suffer from underfitting due to the limited size of the datasets.

V. DISCUSSION

It is observed that the three developed models have different performances. The first model is developed from the depth features alone. The second model is developed from the selected PairCFR depth features. The third model is achieved from the optimized selected pairCFR depth features using the TV selection guide. As shown in Fig. 11, the ensemble classification on the raw depth S_p model may generate more false positives leading to classification errors. Because of the low bias and high variance within the raw depth S_p model. Therefore, an improved S_p model is achieved using PairCFR

TABLE 11. Different depth features learning for various IDF-Sign model.

Features	No. of features	Methods	Iteration	Train time (m/s)	Test time (m/s)	Accuracy (%)	Batch_size
Raw depth	168	Ensemble Tree	120	14290	850	89.55	100
Ranked depth	41	PairCFR	100	800	65	92.77	100
Scaled rank depth	41	IDF-Sign	100	800	45	95.21	100

ranking, where features are ranked according to their weights. The S_p-based PairCFR model improved the recognition performance up to 3%. It is noticed that some features were not determined effectively, because of the nature of decision boundaries by the classification algorithms. We improved the PairCFR capability using the threshold value selection (TV) from the TSA method. The newly obtained TV values improved the recognition performance of S_p up to 6%.

The recognition performance of these algorithms is limited by the high variance and low bias during the training process. To overcome this problem, we apply bagging-based machine learning from an Optimized forest to asses the PairCFR with TV scores. The bagging is introduced in the OF using the decision-based scheme to introduce diversity in the training process, thereby decreasing the level of variance and low bias (over-fitting). The resulting ROC curve is evaluated from the recognized similar feature frequency, however, the discriminant score is employed to read the number of similar features present. The recognition results show that the introduced tree-based bagging scheme can choose optimal subforests that are lesser than the actual forests at the expense of best recognition results than the actual forests.

For signs with recognition scores equal to the TV, TSA substituted the scores with 0.5. We repeated this method to all the considered sign scores. Therefore, a distinct classifier is realized that recognize sign from the same family according to the mid-point score of 0.5 to distinguish true positive rate (TP) of the model from the false positive rate (FP) as shown in Fig. 6. It can be seen that the ROC evaluations determine the threshold of choosing effective recognition scores that may yield the intended depth features. The ROC maps the relationship between the TP of the model and the FP. The ROC produces scores that range from 0.5 to 1 where 1 is the best score and 0.5 means the model is as good as random. We further compute the AUC metrics to determine the proposed IDF-Sign model's true positive rate and the true negative rate, as this metric demonstrate state-of-the-art performance in machine learning. The score ranges from 0% to 100%, where 100% is a perfect score and 0% is the worst. These evaluations are presented in Table 6.

The following sign words return low TP rate; Hear, and Live; whereas Hot, Introduce, Live, Mother, Nephew, Read, and Small, return low ROC, respectively. However, it is observed that the dynamic sign words with low TP rate happen because the information that was utilized as features was not sufficient enough to determine the recognition of these words. We suggest in future research to generate other motion features since these words are highly dynamic in their nature. In addition, dynamic sign words such as Big and Small still exhibit the same class of features that bring wrong recognition results, there is a five-times chance that the dynamic sign word Small appears as Big. The dynamic sign word Read appears six times as Dance. For the problem of wrong recognition because of similar features, we further suggest being addressed by either creating other features at the pixel levels or introducing other methods to reduce the variance of these features.

VI. CONCLUSION

This article is cost-effective because an LMS provides 3D features that are able to transcribe a sign's meaning from simple configuration. The LMS is simple to interface because of non-skin contact surface to interact with the signer. The introduced PairCFR algorithm in the LMS data set leads to achieving low-cost consistent models from the multivariate pairwise-consistency feature ranking of spatial and temporal S_p features. According to the pairwise consistency algorithm, most consistent features in dynamic sign modeling are the hand orientation, velocity, Metacarpophalangeal joints, Proximal joints, and Inter-distal joints in dynamic sign inconsistency modeling. The proposed temporal and spatial information is found very correlated with hand gesture depth indices. The ROC curve analysis was employed to find the spatial-temporal relationship between the effective parameters and the frequency of inconsistency, which is learnt in the ensemble forest classifiers. In the ensemble forest, it is found that optimized forests (OF) demonstrate the best performance. Extensive IDF-Sign model evaluations using Optimized Forest on ASL, GSL-20, and DSG-40 datasets achieved state-of-the-art performance at top-1. The proposed IDF-Sign can be deployed and extended into online sign recognition and translation. In the future, we intend to introduce other computer vision techniques to reduce the variance effect on the depth features.

ACKNOWLEDGMENT

This research is supported by King Mongkut's University of Technology Thonburi's Postdoctoral Fellowship Under Research Project ID 27180. We are also grateful to anonymous IEEE Access reviewers for their potential reviews and insightful comments.

REFERENCES

[1] 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, p. 1406, Feb. 2022.

[2] E.-J. Ong, H. Cooper, N. Pugeault, and R. Bowden, "Sign language recognition using Sequential Pattern Trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2200–2207.

[3] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," in *Gesture Recognition* (The Springer Series on Challenges in Machine Learning), vol. 13. Cham, Switzerland: Springer, 2014, pp. 89–118.

[4] S. B. Abdullahi and K. Chamnongthai, "American sign language words recognition using spatio-temporal prosodic and angle features: A sequential learning approach," *IEEE Access*, vol. 10, pp. 15911–15923, 2022.

[5] S. B. Abdullahi and K. Chamnongthai, "Intelligent fuzzy network for dynamic sign words recognition from spatial features," in *Proc. 19th Int. Conf. Electr. Eng./electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, May 2022, pp. 1–4.

[6] J. Han, G. Awad, and A. Sutherland, "Modelling and segmenting subunits for sign language recognition based on hand motion analysis," *Pattern Recognit. Lett.*, vol. 30, no. 6, pp. 623–633, Apr. 2009.

[7] M. M. Zaki and S. I. Shaheen, "Sign language recognition using a combination of new vision based features," *Pattern Recognit. Lett.*, vol. 32, no. 4, pp. 572–577, Mar. 2011.

[8] N. B. Ibrahim, M. M. Selim, and H. H. Zayed, "An automatic arabic sign language recognition system (ArSLRS)," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 4, pp. 470–477, Oct. 2018.

[9] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 1975–1979.

[10] S. Igari and N. Fukumura, "Sign language word recognition using via-point information and correlation of they bimanual movements," in *Proc. Int. Conf. Adv. Inform., Concept, Theory Appl. (ICAICTA)*, Aug. 2014, pp. 75–80.

[11] J. Simmering, S. M. Z. Borgsen, S. Wachsmuth, and A. Al-Hamadi, "Combining static and dynamic predictions of transfer points for human initiated handovers," in *Proc. Int. Conf. Social Robot.* Cham, Switzerland: Springer, 2019, pp. 676–686.

[12] K. M. Lim, A. W. C. Tan, and S. C. Tan, "A feature covariance matrix with serial particle filter for isolated sign language recognition," *Exp. Syst. Appl.*, vol. 54, pp. 208–218, Jul. 2016.

[13] Q. D. Smedt, H. Wannous, and J.-P. Vandeborre, "3D hand gesture recognition by analysing set-of-joints trajectories," in *Proc. Int. Workshop Understand. Hum. Activities Through 3D Sensors*. Cham, Switzerland: Springer, 2016, pp. 86–97.

[14] S. Y. Boulahia, E. Anquetil, F. Multon, and R. Kulpa, "Dynamic hand gesture recognition based on 3D pattern assembled trajectories," in *Proc. 7th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2017, pp. 1–6.

[15] S. G. Azar and H. Seyedarabi, "Trajectory-based recognition of dynamic Persian sign language using hidden Markov model," *Comput. Speech Lang.*, vol. 61, May 2020, Art. no. 101053.

[16] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos, "Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 249–263.

[17] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, "Decoupled representation learning for skeleton-based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5750–5759.

[18] Y.-B. Li, X.-L. Shen, and S.-S. Bei, "Real-time tracking method for moving target based on an improved Camshift algorithm," in *Proc. Int. Conf. Mech. Sci., Electr. Eng. Comput. (MEC)*, Aug. 2011, pp. 978–981.

[19] P. P. Roy, P. Kumar, and B.-G. Kim, "An efficient sign language recognition (SLR) system using Camshift tracker and Hidden Markov model (HMM)," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–15, Apr. 2021.

[20] Y. Li, D. Ma, Y. Yu, G. Wei, and Y. Zhou, "Compact joints encoding for skeleton-based dynamic hand gesture recognition," *Comput. Graph.*, vol. 97, pp. 191–199, Jun. 2021.

[21] J. Galván-Ruiz, C. M. Travieso-González, A. Pinan-Roescher, and J. B. Alonso-Hernández, "Robust identification system for Spanish sign language based on three-dimensional frame information," *Sensors*, vol. 23, no. 1, p. 481, Jan. 2023.

[22] R. Barioul and O. Kanoun, "K-tournament grasshopper extreme learner for FMG-based gesture recognition," *Sensors*, vol. 23, no. 3, p. 1096, Jan. 2023.

[23] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier," *Exp. Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 118914.

[24] V. Belissen, A. Braffort, and M. Gouiffès, "Dicta-sign-LSF-v2: Remake of a continuous French sign language dialogue corpus and a first baseline for automatic sign language processing," in *Proc. 12th Conf. Lang. Resour. Eval. (LREC)*, May 2020, pp. 6040–6048.

[25] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, Z. T. Al-Qaysi, and A. H. Alamoodi, "Based on wearable sensory device in 3D-printed humanoid: A new real-time sign language recognition system," *Measurement*, vol. 168, Jan. 2021, Art. no. 108431.

[26] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell.*, vol. 151, nos. 1–2, pp. 155–176, Dec. 2003.

[27] F. Jiménez, G. Sánchez, J. Palma, L. Miralles-Pechuán, and J. A. Botía, "Multivariate feature ranking with high-dimensional data for Classification tasks," *IEEE Access*, vol. 10, pp. 60421–60437, 2022.

[28] T. S. Dias, J. J. A. Mendes, and S. F. Pichorim, "Comparison between handcraft feature extraction and methods based on Recurrent Neural Network models for gesture recognition by instrumented gloves: A case for Brazilian Sign Language Alphabet," *Biomed. Signal Process. Control*, vol. 80, Feb. 2023, Art. no. 104201.

[29] M. J. Hussain, A. Shaoor, S. A. Alsuhibany, Y. Y. Ghadi, T. A. Shloul, A. Jalal, and J. Park, "Intelligent sign language recognition system for e-learning context," *Comput., Mater. continua*, vol. 72, no. 3, pp. 5327–5343, 2022.

[30] M. N. Adnan and M. Z. Islam, "Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm," *Knowl.-Based Syst.*, vol. 110, pp. 86–97, Oct. 2016.

[31] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[32] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[33] R. Balamurugan, A. M. Natarajan, and K. Premalatha, "Stellar-mass black hole optimization for biclustering microarray gene expression data," *Appl. Artif. Intell.*, vol. 29, no. 4, pp. 353–381, Apr. 2015.

[34] T. Hanke, "Sign language notation system," in *Proc. 4th Lang. Resour. Eval. Conf. (LREC)*, vol. 4, Lisbon, Portugal, May 2004, pp. 1–6.

[35] S. Tornay, O. Aran, and M. M. Doss, "An HMM approach with inherent model selection for sign language and gesture recognition," in *Proc. 12th Lang. resour. Eval. Conf.*, May 2020, pp. 6049–6056.

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

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 also a member of IEICE, TESA, ECTI, AIAT, APSIPA, TRS, and EEAAT. He is also the Vice President of the Conference of APSIPA Association (2020–2023). He has served as an Editor for *ECTI E-Magazine*, from 2011 to 2015, and an Associate Editor for *ECTI-EEC Transactions*, from 2003 to 2010, and *ECTI-CIT Transactions*, from 2011 to 2016. 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.