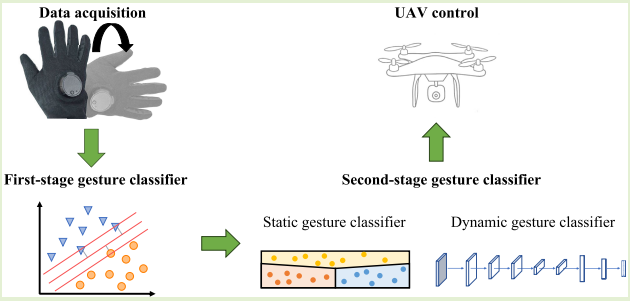
A Two-Stage Real-Time Gesture Recognition Framework for UAV Control

Buyuan Zhang, Haoyang Zhang, Tao Zhen, Bowen Ji, *Member, IEEE*, Liang Xie, Ye Yan, and Erwei Yin

***Abstract*—Unmanned aerial vehicle (UAV) has been widely used in various fields. Traditional UAV controllers require much experience, whereas the control method based on gesture recognition has the advantages of simplicity and flexibility. However, gestures are often simply recognized by current deep learning algorithms, and the static and dynamic properties of gestures are liable to be overlooked, which affects the efficiency of gesture recognition. Hence, a two-stage real-time gesture recognition framework based on the differentiation between static gestures and dynamic gestures is proposed, and gestures toward real-world UAV control are accurately recognized in real time. Besides, a fast correlation-based filter (FCBF) is used to acquire the optimal features. Fifteen gestures, including three static gestures and 12 dynamic gestures, are defined to evaluate the performance of our framework. A practical data glove is meticulously designed with multiple inertial measurement units (IMUs) to obtain the gesture data. Experimental results show that the two-stage framework with FCBF achieves an accuracy of 98.27% under cross-subject cross-validation, outperforming other methods. This work proves the feasibility of optimizing the gesture recognition method by studying the static and dynamic properties of gestures, expecting to facilitate the development of human–computer interaction.**



***Index Terms*— Data glove, feature selection, gesture recognition, inertial measurement unit (IMU), unmanned aerial vehicle (UAV) control.**

I. Introduction

With the rapid development of computer technology and broad application of robotics, human–computer interaction (HCI) is increasingly in demand [1]. HCI is now applied in the fields of medicine, sports, intelligent driving, and so on [2], [3], [4]. Gesture, one of the strong means of expressing emotions and intentions, has the advantages of flexibility and convenience [5]. Gesture recognition has also been favored in recent research as an advantageous approach in HCI with great potential [6].

Unmanned aerial vehicle (UAV) is widely used in telecommunication, rescue operations, and other fields [7]. Although traditional UAV controllers are reliable based on radio technology [8], the learning cost is high, and the portability is not in keeping with the future trend of HCI. Thus, the improvement of interaction experience for UAV control has drawn much more attention. For example, Yu et al. [9] utilized the combination of inertial and bending sensors to recognize gestures for UAV control. The gestures are classified by recurrent neural network (RNN) and the corresponding controlling commands are sent to the UAV. Lu et al. [10] designed a cascading structure to classify gestures using inertial measurement units (IMUs) for UAV control. Gao et al. [11] realize hand gesture recognition to control the UAV by extracting features of surface electromyography (sEMG) signals based on wearable sensors using a multilayer convolutional neural network (CNN) structure.

Based on the data sources of gesture, the gesture recognition methods are mainly classified into two categories, which are vision-based method and wearable sensor-based method. The sources for vision-based gesture recognition methods contain monocular cameras [12], [13], multicameras [14], [15], and

Manuscript received 3 June 2024; accepted 8 June 2024. Date of publication 19 June 2024; date of current version 1 August 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF1203900 and Grant 2023YFF1203903 and in part by the National Natural Science Foundation of China under Grant 62332019 and Grant 62076250. The associate editor coordinating the review of this article and approving it for publication was Dr. Sarbajit Paul. *(Corresponding authors: Haoyang Zhang; Erwei Yin.)*

Buyuan Zhang, Haoyang Zhang, Tao Zhen, Liang Xie, Ye Yan, and Erwei Yin are with the Defense Innovation Institute, Academy of Military Sciences (AMS), Beijing 100071, China, and also with the Intelligent Game and Decision Laboratory, Beijing 100071, China (e-mail: byzhcn@gmail.com; haoyang@tju.edu.cn; zhentao@bjfu.edu.cn; xielnudt@gmail.com; yynudt@126.com; yinerwei1985@gmail.com).

Bowen Ji is with the Unmanned System Research Institute, National Key Laboratory of Unmanned Aerial Vehicle Technology, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: bwji@nwpu.edu.cn).

Digital Object Identifier 10.1109/JSEN.2024.3413787

depth cameras [16], [17], [18], whereas wearable sensor-based gesture recognition methods involve pressure sensors [19], [20], [21], [22], [23], electromyography sensors [24], [25], [26], [27], [28], [29], and inertial sensors [30], [31], [32], [33], [34], [35], which are more flexible and convenient without limitation in terms of the field of view.

Gesture recognition algorithms are normally based on template matching, machine learning, and deep learning. The template matching-based gesture recognition method is dynamic time warping (DTW) [30], [33]. Liu et al. [30] proposed a new template generation method based on DTW for gesture recognition using inertial sensors. The experimental results show that the proposed algorithm for gesture has better movement signal recognition accuracy than existing methods. However, DTW is highly time-consuming to find the optimal solution in large datasets, resulting in high computational cost.

Machine learning-based gesture recognition methods mainly include linear discriminant analysis (LDA) [20], [26], [28] and support vector machine (SVM) [20], [21], [23], [24], [25], [26], [27]. Duan et al. [28] constructed a gesture recognition system based on sEMG signals. More gestures were classified based on LDA by reducing the channels of sEMG. The average accuracy is 91.7%. Shull et al. [20] used ten barometric pressure sensors to recognize gestures, classifying gestures by a modified wristband. The overall accuracy is 94%.

Deep learning-based gesture recognition methods normally consider RNN [19], [36] and CNN [29], [34]. Zhang et al. [19] used IMUs, EMG, and pressure data, employing a long short-term memory (LSTM) algorithm as the gesture classification model, achieving a total accuracy of 89.28% for five dynamic gestures. Wang et al. [34] proposed a sign language recognition system based on IMU and EMG signals, which uses multichannel CNN for gesture classification with an average word error rate of 10.8%.

Machine learning-based methods have relatively weak generalization ability, and they are not able to recognize new gestures effectively. Despite the fact that the generalization ability can be improved by deep learning-based approaches, only different categories of gesture data are directly considered for network training and subsequent recognition, whereas the properties of gestures, such as static gestures and dynamic gestures, are unrecognized. Consequently, the accuracy and speed of gesture recognition can be affected, as presented by nonhierarchical models such as Zhang et al. [19], Maragliulo et al. [27], and Duan et al. [28]. To address the problem, Zhang et al. [21] proposed a gesture recognition method based on the differentiation between static and dynamic gestures. Pan et al. [24] proposed a hierarchical recognition framework based on the predistinction between large motion gestures and subtle motion gestures.

Although the above methods have achieved good performance, challenges still exist in gesture recognition methods for UAV control.

1) Most gesture recognition methods do not focus on the properties of gestures. They directly utilize classifiers for recognition, which can cause unsatisfactory results.
2) Though the hierarchical framework considers the properties of gestures, current classifiers used in the hierarchical framework have limited generalization ability to satisfy the demand of recognizing complex dynamic gestures.
3) Current feature selection methods are mainly based on empirical selection of the sensor type, instead of the characteristics of data, which affects the performance of the model.
4) Online recognition algorithms generally use the sliding window with fixed step size, which cannot adjust to the characteristics of gestures well. For some simple static gestures, overlapping windows with small step sizes contain duplicate gesture information, leading to an increase in computational cost.

Our main contributions are summarized as follows.

1) We propose a two-stage gesture recognition framework, which differentiates static and dynamic gestures using machine learning-based classifiers in the first stage, and then recognized specific gestures in the second stage. We first formally introduce the idea of considering the static and dynamic properties of gestures in the hierarchical framework, which improves the accuracy of traditional gesture recognition methods.
2) The limitations of hierarchical frameworks are optimized for dynamic gesture recognition. We combine CNN and LSTM algorithms to reduce the recognition cost associated with DTW and improve the generalization ability of SVM and deep belief networks (DBNs), thereby enhancing the robustness of traditional hierarchical frameworks.
3) The fast correlation-based filter (FCBF) is utilized to select the best subset of features. It ensures that the values of information gain (IG) are comparable, thereby reducing the dimension of the features and improving the recognition speed of the two-stage model.
4) We propose a novel adaptive step-based sliding window method for online gesture recognition that realizes real-time classification precisely and portable control of UAVs.

The rest of this article is organized as follows. Section II describes the experimental environment and the details of our proposed gesture recognition algorithm. Section III presents the experimental results. Section IV discusses the proposed method and its limitations. Finally, conclusions are presented in Section V.

II. Methods

A. Overview of the Proposed System

The two-stage classification framework proposed in this article consists of raw signal stream, data preprocessing, feature extraction, feature selection, first-stage gesture classifier, and second-stage gesture classifier. Raw data of gestures acquired by IMU data gloves are transmitted via Bluetooth or Universal Serial Bus (USB) to be filtered, normalized, and detected by the data preprocessing. Afterwards, statistical and frequency domain features are extracted and selected to improve the efficiency of the model. The first-stage gesture classifier differentiates static gestures and dynamic gestures, which are then separately forwarded to the corresponding

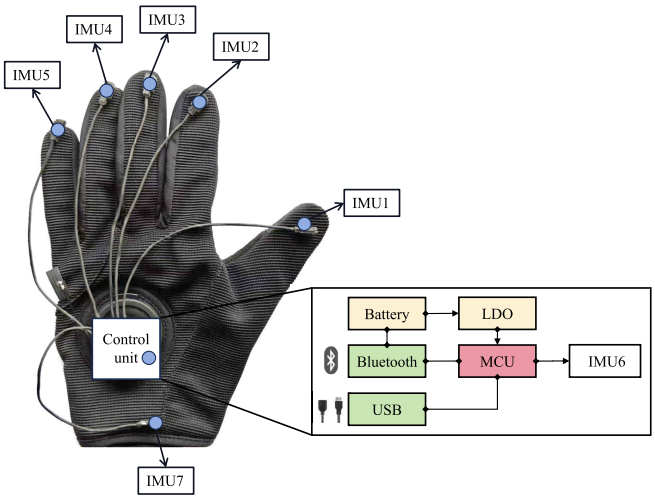


Fig. 1. Hardware structure of the data glove.

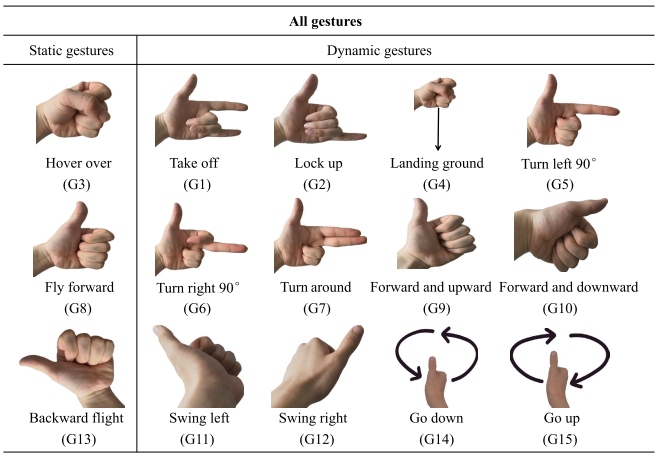


Fig. 2. All gestures of the UAV control system, including static and dynamic gestures.

TABLE I
LIST OF FEATURES FOR ACCELEROMETER AND GYROSCOPE SIGNALS

Feature name	Dimension
Mean	1
Variance	1
Max	1
Min	1
Standard deviation	1
Integration	1
Mean absolute deviation	1
Root mean square	1
Zero cross rate	1
Skewness	1
Kurtosis	1
First five orders of 256-point FFT coefficients	5
Entropy	1
Signal magnitude area	1
10 order AR coefficients	10
Average power spectral density	1
Peak power spectral density	1
Mean absolute value	1
Wave-length	1
Total	32

classifiers in the second stage. Static and dynamic gestures are finally recognized by the machine learning classifier and neural network in the second-stage gesture classifier, respectively.

B. System Hardware and Software

In our system, seven nine-axis motion tracking sensors were selected to characterize the hand motion. We developed a practical data glove integrating seven sensors while fully considering the industrial design and interactive experience. Fig. 1 shows the hardware structure of the data glove, with five IMU (IMU1–IMU5) being located between the distal and middle phalanges of the fingers, IMU6 on the center at the back of the hand, and IMU7 at the wrist, sampling the hand motion at 100 Hz. In the control unit, low dropout regulator (LDO) is a voltage regulator that provides power to other modules. Microcontroller unit (MCU) is a microcontroller that reads and writes the values of the registers. The data acquisition interface of the data glove is developed using PyCharm and PyQt5 software development kits.

C. Data Acquisition and Preprocessing

As shown in Fig. 2, a total of 15 gestures, including three static and 12 dynamic categories, are designed for UAV control. Ten healthy subjects (seven males/three females, 24.6 $\pm$ 2.6 years old) have participated in this study, and each gesture was performed 20 times by each subject, resulting in a dataset of 3000 (15 $\times$ 10 $\times$ 20) gestures. A visual prompt is designed for the start of each gesture during the data acquisition. When a gesture has been performed, the participant is informed by a timer to take a break before the following acquisition.

After data acquisition, a low-pass filter is used to reduce noise, and the Tustin's method is applied to transform the analog low-pass filter into its digital counterpart. The output of the Nth-order infinite impulse response filter is given by

$$y_n = \sum_{i=0}^{M} b_i x_{n-i} - \sum_{i=1}^{N} a_i y_{n-i} \tag{1}$$

where b_i and a_i are obtained by converting the poles and zeros into the coefficients b and a of the polynomial function by dividing the first coefficient a_0. x is the input and y is the output. Then, the filtering process is completed. Fig. 3 shows the comparison of the accelerometer and gyroscope signals before and after filtering in terms of static gesture G13 and dynamic gestures G1, G7, and G11. Notice that after filtering, most of the noise in the raw signal can be eliminated. It somewhat avoids the effect of noise on the gesture recognition accuracy.

In order to eliminate the effect of magnitude differences on the gesture recognition results, the filtered signals are normalized between $(-1, 1)$. The normalization process is given by

$$N(S) = -1 + \frac{2(S - S_{\min})}{S_{\max} - S_{\min}} \tag{2}$$

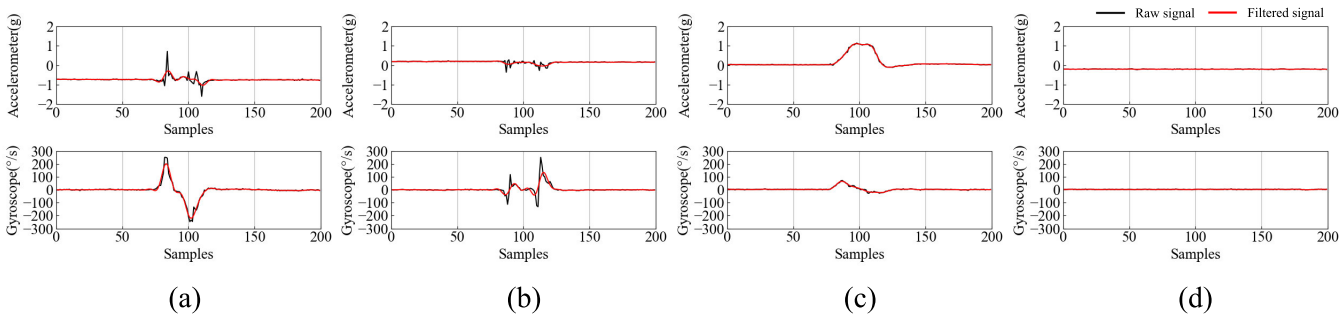


Fig. 3. Comparison of single-channel accelerometer and gyroscope signals before and after filtering for four categories of gestures, respectively. (a) Gesture G1. (b) Gesture G7. (c) Gesture G11. (d) Gesture G13. Here, (a)–(c) are dynamic gestures and (d) is a static gesture. The top of the subfigure shows the comparison of accelerometer signals before and after filtering for each gesture, and the bottom of the subfigure shows the comparison of gyroscope signals before and after filtering for each gesture.

where S is the original signal, $S_{\min}$ and $S_{\max}$ are the minimum and maximum values of the original signal, respectively, and $N(S)$ is the result after signal normalization.

Finally, active segments need to be identified from the normalized signal stream for gesture classification. In this article, we propose an energy calculation method utilizing adaptive thresholding to determine the threshold value for active segment detection, which is calculated as

$$E(i) = \sum_{j=1}^{C} G_{ij}^2 \tag{3}$$

$$\theta = \alpha(E_{\max} - E_{\min}) \tag{4}$$

where G is the value of gyroscope, C is the number of channels of the gyroscope, E is the sequence of energy values, $E_{\max}$ is the maximum energy value, $E_{\min}$ is the minimum energy value, α is the proportionality constant (in this article, we take 0.14), and θ is the threshold value.

D. Feature Extraction

We consider that the raw data may have redundancy. Feature extraction can reduce the amount of computation and better reflect the characteristics of the signals. The statistical and frequency-domain features extracted from the accelerometer and gyroscope signals in this article are shown in Table I, which have been widely used in related studies [24], [37], [38]. The features were extracted from three-axis accelerometers and three-axis gyroscopes, with 19 features extracted per axis for a total of 32 values. As a result, the feature dimension extracted from one single IMU is 6×32, representing a fourfold reduction compared to the original data dimension of 6×200.

E. Feature Selection

Feature selection is typically used to combine the features that contribute significantly to the classification task into the best subset of features. It helps to reduce the feature dimensions, eliminate redundant data, and improve the efficiency of the algorithm.

There are three typical feature selection methods, which are filter methods, wrapper methods, and embedded methods [37]. Filter methods select features by the correlation metric between features and category labels that are independent of the target classifier. While wrapper methods and embedded methods evaluate features based on their performance within the classifier, making their selection criteria being directly related to the target classifier. Therefore, in this work, the FCBF algorithm [39] of filter methods is chosen for feature selection. Initially, this algorithm calculates the IG between features, and the class labels are given by

$$\mathrm{IG}(X|Y) = H(X) - H(X|Y) \tag{5}$$

where $\mathrm{IG}(X|Y)$ is the IG, and $H(X)$ is the entropy of variable X, defined as

$$H(X) = -\sum_{i} P(x_i)\log_2((P(x_i))) \tag{6}$$

where $P(x_i)$ is the prior probabilities for all values of X. $H(X|Y)$ is the entropy of variable X under the condition of observing variable Y, defined as

$$H(X|Y) = -\sum_{i} P\left(y_j\right) \sum_{i} P\left(x_i|y_j\right)\log_2\left(P\left(x_i|y_j\right)\right) \tag{7}$$

where $P(y_i)$ is the prior probabilities for all values of Y. $P(x_i|y_j)$ is the posterior probabilities of X given the values of Y.

The FCBF algorithm introduces the concept of symmetrical uncertainty (SU) to normalize the IG, thus eliminating the scale effect and rendering the IG comparable. SU is defined as

$$\mathrm{SU}(X, Y) = 2\left[\frac{\mathrm{IG}(X|Y)}{H(X) + H(Y)}\right] \tag{8}$$

where $\mathrm{SU}(X, Y)$ is the SU, and $H(Y)$ is the entropy of variable Y. For a given dataset of N features and a class C, the FCBF method first calculates the SU value for each feature. Relevant features are selected into S_{list} based on the predefined threshold δ. They are sorted in descending order according to their SU values. Then, select P as the dominant feature and find the next feature Q of P in S_{list}. If $\mathrm{SU}(P, Q) \geq \mathrm{SU}(Q, C)$, it means the feature Q is the redundant feature (highly correlated between features and uncorrelated between feature and class), which is then removed from S_{list}. Finally, the predominant features preserved are the optimal subset.

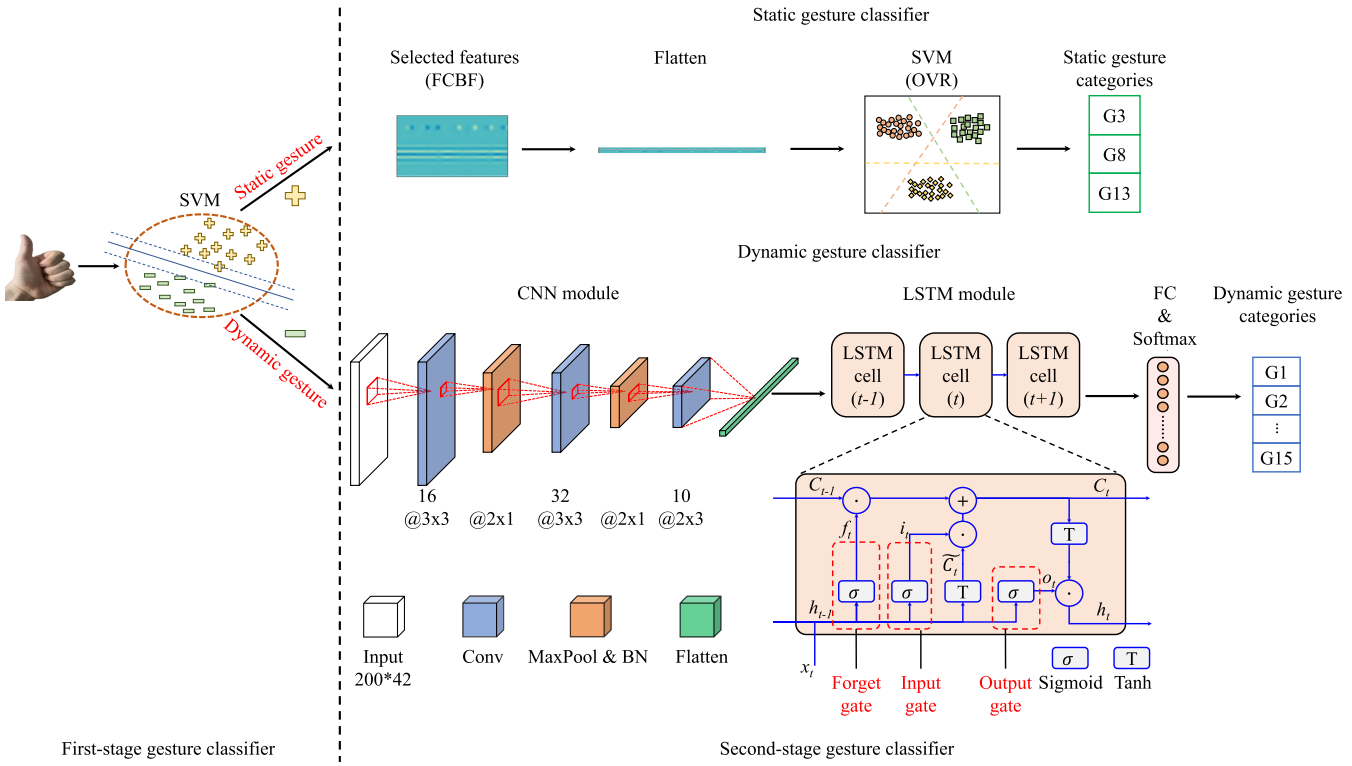


Fig. 4. Structure of two-stage gesture classifiers.

	Static gestures			Dynamic gestures										
Gestures														
Label No.	0	1	2	0	1	2	3	4	5	6	7	8	9	10	11
Category No.	G3	G8	G13	G1	G2	G4	G5	G6	G7	G9	G10	G11	G12	G14	G15

Fig. 5. Mapping of labels for static and dynamic gestures.

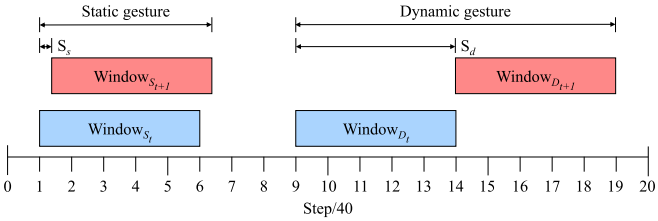


Fig. 6. Example of the adaptive step sliding window.

F. Architecture of the Two-Stage Framework

The architecture of the two-stage gesture recognition framework is shown in Fig. 4. It is divided into a first-stage classifier, which distinguishes static and dynamic gestures, and a second-stage classifier, which determines the exact static or dynamic gesture.

In the first stage, each gesture is given a static or dynamic label according to the motion characteristics. The gesture data and labels are then used to train classifiers such as SVM to recognize specific static and dynamic gestures. If the classification result is static, then the gesture data are forwarded to the static gesture classifier in the second stage for final classification. Similarly, if the gesture is classified as a dynamic one, the gesture data are directed to the dynamic gesture classifier in the second stage for final classification.

The second-stage classifiers are composed of a static gesture classifier and a dynamic gesture classifier. The former uses the FCBF algorithm to select features, which are flattened by dimension transformation. Then, a classifier such as SVM recognizes gestures through the 1-D features. The latter selects CNN to extract features and classifies time series by LSTM.

G. First-Stage Gesture Classifier

This article considers traditional machine learning classifiers, including linear regression (LR), k-nearest neighbor (KNN), decision tree (DT), SVM, and Naïve Bayes (NB) to classify gestures. Note that the classification here is to differentiate static and dynamic gestures. SVM serves as the first-stage classifier, as shown in Fig. 4. It can construct a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.

In the case where the samples are not linearly divisible, SVM divides the hyperplane in the feature space corresponding to the model as

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) + b \tag{9}$$

where $\mathbf{w}$ is the normal vector to the hyperplane, $\phi(\mathbf{x})$ maps $\mathbf{x}_i$ to a higher dimensional space, $\mathbf{x}_i$ is the training vector, and b is the bias.

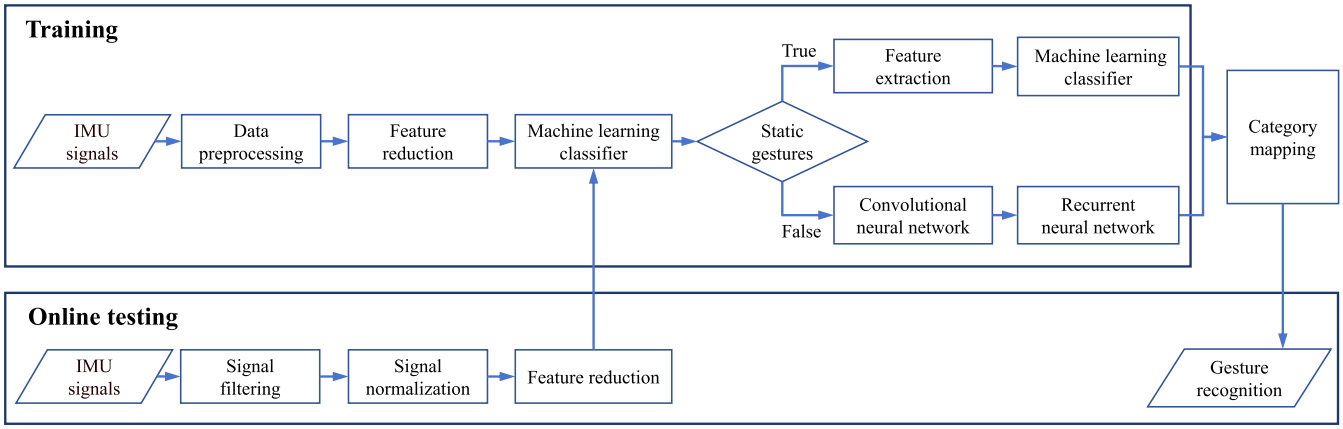


Fig. 7. Flowchart of the algorithm for gesture recognition.

The optimization problem for SVM with the introduction of slack variable ξ at soft margin is

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{s.t. } y_i\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i)+b\right) \geq 1-\xi_i, \quad \xi_i \geq 0 \quad \forall i \in \{1,\ldots,n\} \tag{10}$$

where C is the regularization parameter, n is the size of sample set, and y_i is the class of the sample. From (10), we obtain the following dual problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \kappa\left(\mathbf{x}_i,\mathbf{x}_j\right)$$
$$\text{s.t. } \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \forall i \in \{1,\ldots,n\} \tag{11}$$

where α is the Lagrange multiplier, and $\kappa(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^{\mathrm{T}}\phi(\mathbf{x}_j)$ is the kernel function, in order to avoid computing high-dimensional features of inner products directly. The dual problem is solved for $\mathbf{w}$ as

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i). \tag{12}$$

Substituting $\mathbf{w}$ and $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ into $f(\mathbf{x})$ yields the following equation:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x},\mathbf{x}_i) + b \tag{13}$$

where $f(\mathbf{x})$ is the solution of the hyperplane. SVM uses the derived hyperplane to classify the gestures.

Algorithm 1 Adaptive Sliding Window Algorithm

Input: $\mathbf{d}$ data stream
w window size
δ threshold of dynamic gesture probability
S_s static gesture step
S_d dynamic gesture step
Output: $\mathbf{G}$ gesture sequence

$i \leftarrow 0$
$j \leftarrow 1$
$adjustFlag \leftarrow$ False
while $i + w \leq$ LENGTH$\mathbf{d}$ **do** ▷ Read data stream.
  $\mathbf{d}_w \leftarrow \mathbf{d}[i : i + w]$
  $g_c, g_p \leftarrow$ PREDICT$(\mathbf{d}_w)$ ▷ Recognize gestures.
  if g_c is static **then**
    $\mathbf{G}(j) \leftarrow g_c$
  else if g_c is dynamic and $g_p \geq \delta$ **then**
    $\mathbf{G}(j) \leftarrow g_c$
    $adjustFlag \leftarrow$ True
  else
    $\mathbf{G}(j) \leftarrow -1$ ▷ Mark illegal gestures.
    goto 16
  end if
  if $adjustFlag$ is True **then** ▷ Adjust the step.
    $i \leftarrow i + S_d$
    $adjustFlag \leftarrow$ False
  else
    $i \leftarrow i + S_s$
  end if
  $j \leftarrow j + 1$
end while

H. Second-Stage Gesture Classifier

1) Static Gesture Classifier: The structure of the static gesture classifier in the second stage is shown in Fig. 4. It performs the final classification of the three static gestures. Similarly, traditional machine learning classifiers such as LR, KNN, DT, SVM, and NB are considered. In addition, the training data of the static gesture classifier in the second stage have been repartitioned, which results in the change of labels, as shown in Fig. 5.

2) Dynamic Gesture Classifier: The dynamic gesture classifier performs the final classification of the 12 dynamic gestures. CNN and LSTM are combined in the dynamic gesture classifier. As shown in Fig. 4, raw data in the shape of 200 × 42 are input to CNN to extract features. Features are input to LSTM for final prediction. The implementation formulas of LSTM are as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{14}$$

TABLE II
TEN FEATURES SELECTED BY FCBF FEATURE SELECTION METHOD

Rank	Feature name	Dimension
#1	10 order AR coefficients	10
#2	First five orders of 256-point FFT coefficients	5
#3	Variance	1
#4	Mean absolute value	1
#5	Max	1
#6	Mean	1
#7	Root mean square	1
#8	Wave-length	1
#9	Standard deviation	1
#10	Entropy	1

TABLE III
COMPARISON OF FCBF AND NONFCBF FEATURE SELECTION METHODS (MEAN ± STANDARD DEVIATION (SD))

Model	Accuracy		Time	
	FCBF	NonFCBF	FCBF	NonFCBF
LR	98.50±2.99	99.63±0.51	91.63±3.37	153.09±5.93
KNN	98.23±1.50	98.07±1.62	131.62±6.39	196.44±8.45
DT	99.77±0.40	99.23±1.09	90.91±2.45	152.53±5.63
SVM	99.33±1.35	99.77±0.34	90.99±2.69	152.98±6.16
NB	99.67±0.45	99.00±1.41	90.47±2.63	154.76±6.23

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{15}$$
$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{16}$$
$$\tilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{17}$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{18}$$
$$h_t = o_t \cdot \tanh(C_t). \tag{19}$$

where σ is the sigmoid function, f_t is the forget gate, i_t is the input gate, o_t is the output gate, $\tilde{C}_t$ is the candidate cell state, x_t is the input, W and b are the weight matrix and bias terms, respectively, C_t is the cell state, and h_t is the hidden state.

We selected cross-entropy loss to train the dynamic gesture classifier

$$\mathcal{L}(x, y) = \{l_1, \ldots, l_N\}^{\mathrm{T}}, \quad l_n = -\sum_{c=1}^{C} w_c \log \frac{e^{x_{n,c}}}{\sum_{i=1}^{C} e^{x_{n,i}}} y_{n,c} \tag{20}$$

where $\mathcal{L}$ is the loss function, x is the input, y is the target, N is the batch size, C is the number of classes, and w is the weight. The classification results are mapped to the final categories according to Fig. 5.

3) Online Recognition Algorithm: In this article, an adaptive sliding window with variable steps is proposed to realize online gesture recognition for UAV control. As shown in Fig. 6, if the window recognizes the gesture as static, denoted as Window_{S_t}, the window's step is set to S_s. In contrast, if the window recognizes the gesture as dynamic (Window_{D_t}), the step is set to S_d. Algorithm 1 shows the pseudocode of the adaptive sliding window in detail. Here, the w is set to 200, and the δ is set to 0.99. The S_s and the S_d are set to 10 and 200 samples, respectively.

TABLE IV
FIRST-STAGE RESULTS OF STATIC AND DYNAMIC GESTURE DIFFERENTIATION USING OPTIMAL HYPERPARAMETERS UNDER 10-FOLDCV (MEAN ± SD)

Model	Accuracy	Precision	Recall	F1-score
LR	99.87±0.27	99.83±0.39	99.74±0.52	99.78±0.44
KNN	98.80±0.54	98.09±1.16	98.14±0.77	98.11±0.85
DT	99.67±0.33	99.51±0.54	99.49±0.62	99.50±0.49
SVM	99.90±0.21	99.89±0.27	99.80±0.39	99.85±0.31
NB	99.77±0.26	99.86±0.16	99.38±0.71	99.61±0.44

TABLE V
FIRST-STAGE RESULTS OF STATIC AND DYNAMIC GESTURE DIFFERENTIATION USING OPTIMAL HYPERPARAMETERS UNDER LOPOCV (MEAN ± SD)

Model	Accuracy	Precision	Recall	F1-score
LR	98.50±2.99	97.83±4.91	98.13±3.20	97.81±4.16
KNN	98.23±1.50	98.16±1.79	96.40±3.59	97.13±2.54
DT	99.77±0.40	99.67±0.63	99.60±0.65	99.63±0.62
SVM	99.33±1.35	98.74±2.76	99.40±0.86	99.01±1.94
NB	99.67±0.45	99.79±0.28	99.17±1.12	99.47±0.72

TABLE VI
PARAMETERS OF THE DYNAMIC GESTURE CLASSIFIER (EPOCH = 50, BATCH SIZE = 64, LEARNING RATE = 0.01, OPTIMIZER = ADAM, AND LOSS FUNCTION = CROSS ENTROPY LOSS)

Layer	Parameter
Convolutional layer	kernel_size = 3x3, stride = 1, padding = 1, activation function = ReLU
Batch normalization layer	num_features = 16
Max pooling layer	kernel_size = 2x1, stride = 2x1, padding = 1
Convolutional layer	kernel_size = 3x3, stride = 1, padding = 1, activation function = ReLU
Batch normalization layer	num_features = 32
Max pooling layer	kernel_size = 2x1, stride = 2x1, padding = 1
Convolutional layer	kernel_size = 3x3, stride = 1, padding = 1, activation function = ReLU
Flatten layer	start_dim = 1, end_dim = 2
LSTM layer	input_size = 42, hidden_size = 32, num_layers = 1
Fully connected layer	in_features = 32, out_features = 12, activation function = LogSoftmax

In summary, the algorithmic flowchart is shown in Fig. 7. The whole process can be divided into a training phase and an online testing phase. The former is mainly designed for updating the parameters of the classifier. The latter mainly focuses on recognizing unlabeled gesture data, obtaining gesture classification results, and realizing UAV control. Both phases adopt active segments or sliding windows for the classifier to recognize gestures.

III. RESULTS

We use four evaluation metrics with two evaluation methods to assess the performance of the model. The evaluation metrics are: accuracy, precision, recall, and F1-score, calculated as

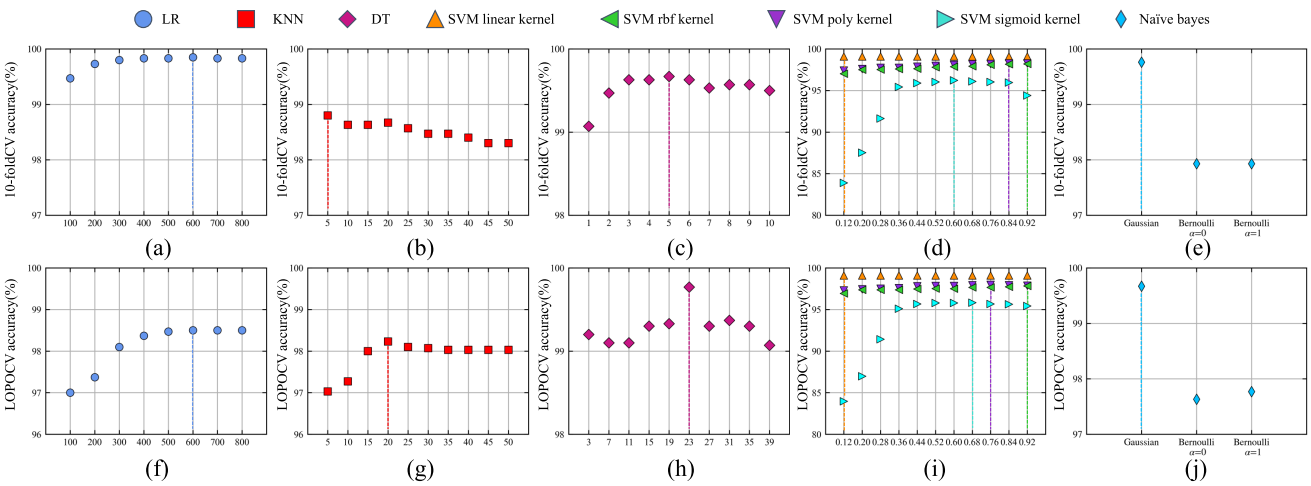


Fig. 8. Results of optimal hyperparameter selection for the first-stage machine learning classifier. (a)–(e) Hyperparameter selection for the maximum number of iterations, number of neighbors, maximum depth, regularization parameter, and model type under 10-foldCV. (f)–(j) Hyperparameter selection for the maximum number of iterations, number of neighbors, maximum depth, regularization parameter, and model type under LOPOCV.

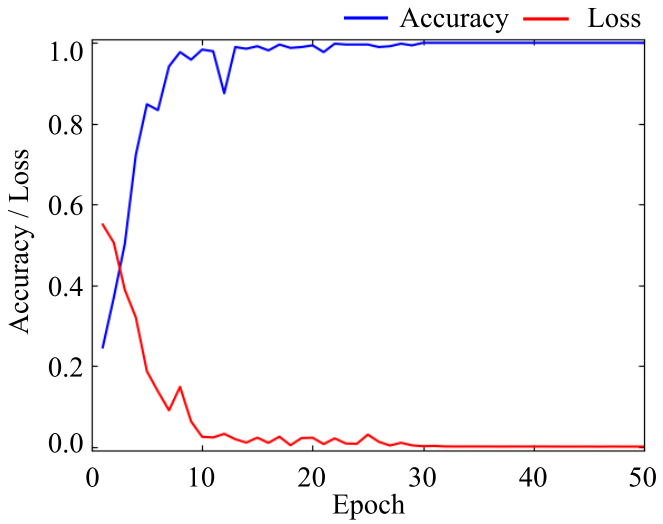


Fig. 9. Training accuracy and loss for the dynamic gesture classifier.

Fig. 10. UAV flight environment in real scenes.

follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{21}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{22}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{24}$$

The evaluation methods are tenfold cross-validation (10-foldCV) and leave-one-participant-out cross-validation (LOPOCV), which are detailed as follows.

1) *10-foldCV:* All the training segments are divided into ten parts; one part is used as a test set and the other nine parts are used as the training set. The process is performed in an iterative way until all the parts have been used as test set.

2) *LOPOCV:* All the training segments are divided into tenfold based on different subjects, and each time the training set consists of gesture segments from nine subjects, and the remaining one is used as the test set. The process is also performed iteratively until all subjects' gesture segments have been used as the test set.

A. Feature Selection

In this work, the FCBF method is applied for feature selection and feature ranking. Table II shows ten selected features, such as ten-order autoregressive (AR) coefficients and first five orders of 256-point fast Fourier transform (FFT) coefficients and variance.

Table III shows the comparison of the accuracy and time performance with and without the FCBF feature selection method. We can see that the FCBF method improves the recognition speed of the model with almost no loss in accuracy. For example, when using SVM, the FCBF method achieves an accuracy of 99.33% in 90.99 ms, whereas the nonFCBF

TABLE VII
COMPARISON OF TWO-STAGE AND NONTWO-STAGE GESTURE CLASSIFICATION USING 10-FOLDCV

Gesture category	Two-stage			Nontwo-stage		
	Precision	Recall	F1-score	Precision	Recall	F1-score
G1	99.50	100.00	99.75	98.02	99.00	98.51
G2	100.00	100.00	100.00	98.98	97.50	98.23
G3	99.01	100.00	99.50	98.52	100.00	99.26
G4	100.00	100.00	100.00	99.00	99.00	99.00
G5	100.00	99.50	99.75	97.97	96.50	97.23
G6	98.51	99.50	99.00	97.07	99.50	98.27
G7	99.49	98.50	98.99	97.43	99.50	96.20
G8	100.00	98.50	99.24	98.50	98.50	98.50
G9	100.00	99.50	99.75	98.98	97.00	97.98
G10	99.50	99.00	99.25	99.50	99.50	99.50
G11	99.00	99.50	99.25	98.49	98.00	98.25
G12	99.50	100.00	99.75	97.54	99.00	98.26
G13	100.00	100.00	100.00	100.00	100.00	100.00
G14	100.00	99.00	99.50	95.38	93.00	94.18
G15	98.52	100.00	99.26	92.31	96.00	94.12
Average	99.54	99.53	99.53	97.85	97.83	97.83
Variance	0.276	0.282	0.116	3.349	3.789	2.846

TABLE VIII
COMPARISON OF TWO-STAGE AND NONTWO-STAGE GESTURES CLASSIFICATION USING LOPOCV

Gesture category	Two-stage			Nontwo-stage		
	Precision	Recall	F1-score	Precision	Recall	F1-score
G1	100.00	100.00	100.00	90.28	97.50	93.75
G2	100.00	100.00	100.00	97.33	91.00	94.06
G3	87.80	90.00	88.89	92.74	89.50	91.09
G4	100.00	100.00	100.00	98.96	95.50	97.20
G5	100.00	100.00	100.00	94.55	95.50	95.02
G6	100.00	100.00	100.00	91.63	98.50	94.94
G7	100.00	100.00	100.00	87.95	98.50	92.92
G8	89.69	87.00	88.32	86.32	91.50	88.83
G9	100.00	97.50	98.73	98.40	92.50	95.36
G10	100.00	100.00	100.00	93.07	94.00	93.53
G11	100.00	100.00	100.00	96.06	97.50	96.77
G12	100.00	100.00	100.00	98.50	98.50	98.50
G13	100.00	100.00	100.00	100.00	100.00	100.00
G14	99.01	100.00	99.50	84.58	85.00	84.79
G15	97.55	99.50	98.51	94.51	77.50	85.16
Average	98.27	98.27	98.26	93.66	93.50	93.46
Variance	14.474	15.362	14.576	21.383	34.200	18.209

method achieves an accuracy of 99.77% in 152.98 ms. The results validate the effectiveness of the FCBF method in feature selection.

B. Optimization of Models and Hyperparameters

Fig. 8 shows the optimal results for hyperparameters of each classifier under 10-foldCV and LOPOCV in the first stage. As shown in Fig. 8(a) and (f), when the LR model is used, the optimal number of iterations is 600 under both 10-foldCV and LOPOCV. As for Fig. 8(b) and (g), when the KNN model is used, the optimal number of neighbors are 5 under 10-foldCV and 20 neighbors under LOPOCV, respectively. From Fig. 8(c) and (h), we can see that when using the DT model, the optimal depths are 5 under 10-foldCV and 23 under LOPOCV, respectively. When the SVM model is used, the optimal kernel under both 10-foldCV and LOPOCV is the linear one, and the optimal regularization parameter is 0.12 [Fig. 8(d) and (i)]. Finally, when the NB model is used, the optimal classifier distribution under both 10-foldCV and LOPOCV is Gaussian as shown in Fig. 8(e) and (j), with an accuracy of 99.77% and 99.67%, respectively.

On the one hand, Table IV shows the results of using optimal hyperparameters in each classifier under 10-foldCV. SVM has the highest accuracy in differentiating static and dynamic gestures. On the other hand, Table V shows the results under LOPOCV, and DT achieves the highest accuracy. We can see that optimizing the hyperparameters of classifiers is effective in improving the framework accuracy.

Statistical analysis shows that there is a significant difference in accuracy between the models ($p = 0.003$, statistical significance is set to $p = 0.05$). Further posthoc analysis reveals significant differences between KNN and both DT or SVM ($p < 0.05$).

The parameters for each layer of the dynamic gesture classifier are listed in Table VI. Fig. 9 shows the accuracy and loss curves of the model during training, which converges rapidly as the epoch increases.

C. Classification Results

The gesture recognition results of both the two-stage model and the nontwo-stage model under 10-foldCV are compared in Table VII. Note that for the nontwo-stage model, we utilize the structure of CNN combined with LSTM, instead of a cascaded structure, to construct the classifier. The average precision and recall of the two-stage model are 99.54% and 99.53%, respectively, which are better than the those of the nontwo-stage model. Besides, the recognition accuracies for G14 and G15 are significantly improved when using the two-stage model.

Table VIII shows the results under LOPOCV. The average precision and recall under the two-stage model are 98.27% and 98.27%, respectively, which are 4.61% and 4.77% higher than those using the nontwo-stage model. For all gestures except for G3, the precision and recall are greatly improved. This can be attributed to the limitation in the generalization ability of classifiers when recognizing static gestures across participants. Nevertheless, the performance consistency for all the evaluation items has been evidently optimized when using the two-stage model.

D. Online Recognition Validation

In terms of online recognition validation, the subject intends to perform UAV control wearing the data glove via the 15 gestures. Each gesture category was tested 20 times for totally 300 gestures. The online recognition accuracy of three static gestures and 12 dynamic gestures reaches 100%. The validation results show that the combination of data gloves and recognition algorithms can precisely recognize gestures.

We conducted the UAV flight control in the real-world test scenario, and the environment is shown in Fig. 10. The test was performed outdoors, and interference from complex conditions

TABLE IX
COMPARISON OF OUR FRAMEWORK WITH THE PREVIOUS METHOD

		This work	Yu *et al.* [9]	Zhang *et al.* [21]	Pan *et al.* [24]	Zhang *et al.* [19]	Maragliulo *et al.* [27]	Duan *et al.* [28]
System	Method	SVM+CNN+LSTM	BP+Bi-GRU	SVM+DTW	SVM+DBN	LSTM	SVM	LDA
	Sensors	IMU	FSR+IMU	FSR	EMG+IMU	EMG+IMU+FSR	EMG	EMG
	Numbers of gestures	15	15	8	21	10	5	9
	Layered framework	√	×	√	√	×	×	×
	Static and dynamic gesture differentiation	√	√	√	×	×	×	×
	Latency per gesture	0.0934 s	0.0086 s	6.8621 s	0.0821 s	0.0081 s	0.0039 s	0.0607 s
Accuracy (10-foldCV / LOPOCV)	Static gestures	99.50% / 92.33%	94.67% / 82.50%	100.00% / 100.00%	89.33% / 68.17%	100.00% / 93.33%	61.67% / 43.83%	91.83% / 85.67%
	Dynamic gestures	99.54% / 99.75%	98.50% / 92.88%	89.92% / 33.42%	93.25% / 86.17%	95.79% / 91.00%	67.46% / 47.17%	96.83% / 91.92%
	Static and dynamic gestures	**99.53% / 98.27%**	97.73% / 90.80%	91.93% / 46.73%	92.47% / 82.57%	96.63% / 91.47%	66.30% / 46.50%	95.83% / 90.67%

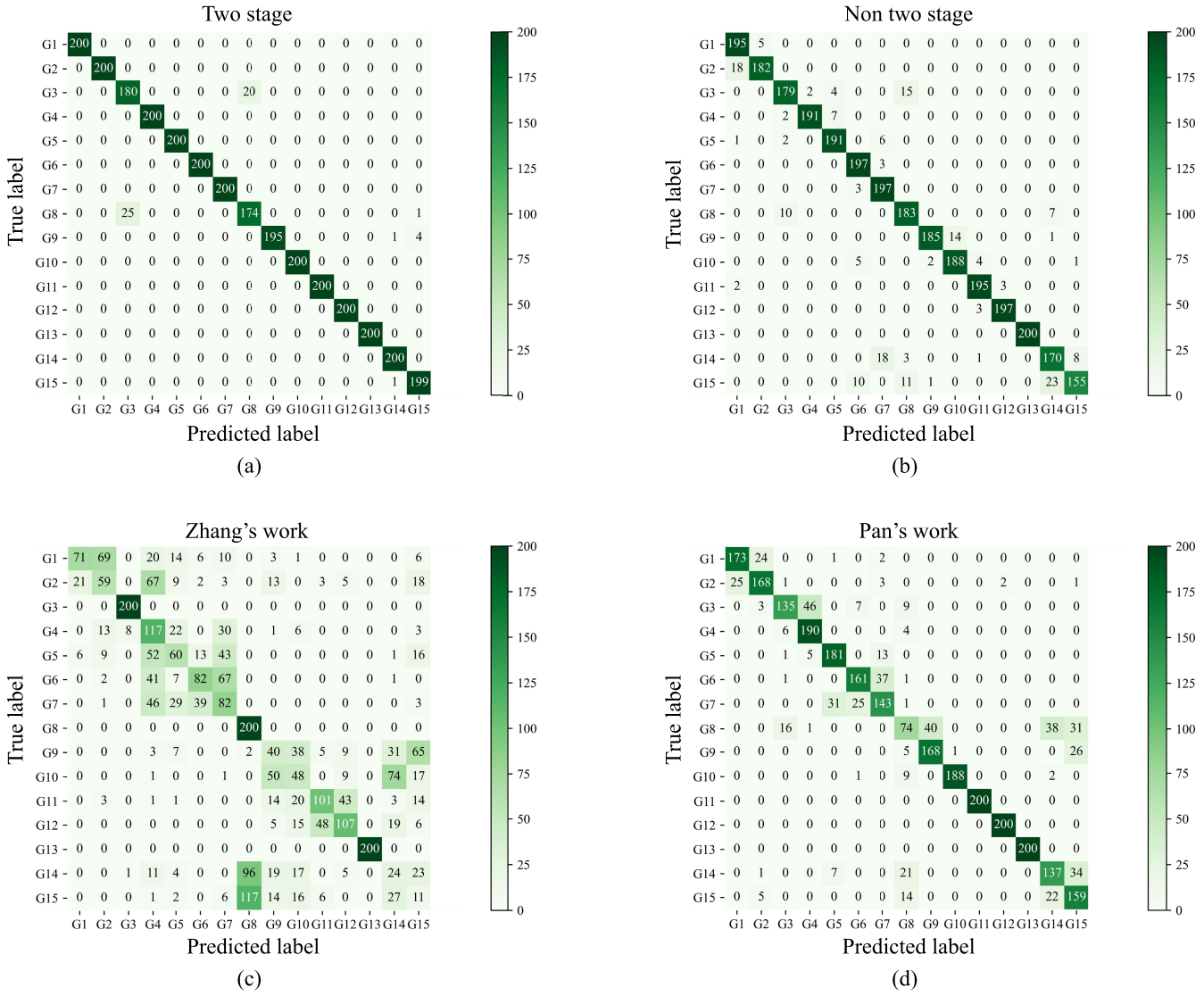


Fig. 11. Confusion matrices of four models for the recognition of 15 gestures. (a) Confusion matrix of the two-stage model proposed in this article. (b) Confusion matrix of the nontwo-stage model. (c) Confusion matrix of Zhang's work. (d) Confusion matrix of Pan's work.

such as tree branches and walls existed at the site. During the flight, there were no occurrences of control failure or collision, and the UAV was able to avoid the obstacles. Furthermore, it proves the effectiveness, reliability, and value of the practical application of our proposed method for UAV control.

E. Comparison With Other Methods

In order to further analyze the advantages of the two-stage model and prove the effectiveness of our improvement on the traditional hierarchical model, the methods used in [9], [21], and [24] are selected for comparison, which is shown in

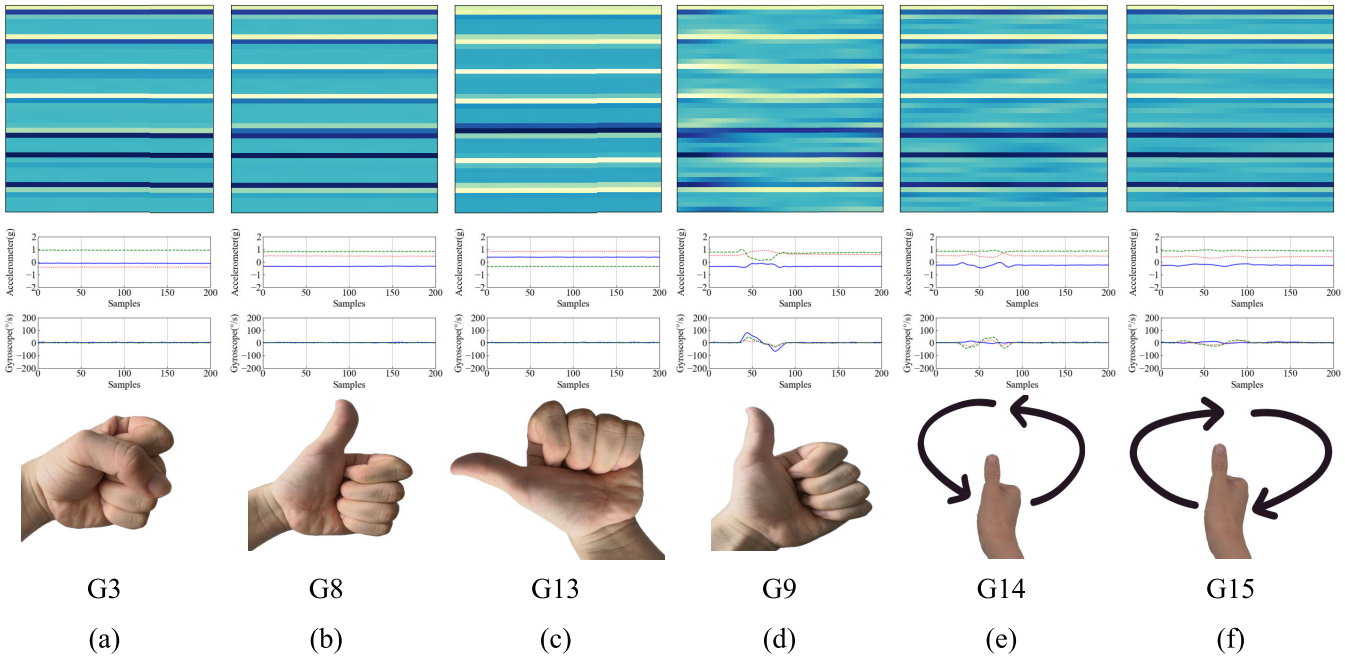


Fig. 12. Signal visualization for six gestures. (a)–(f) Signal visualization of G3, G8, G13, G9, G14, and G15, with the top line of subfigures showing the signal of 42 channels (seven IMUs × three channels of accelerometer and seven IMUs × three channels of gyroscope) of gesture segments, each of the six rows corresponds to the three-axis accelerometer and three-axis gyroscope data for one IMU, the order of each IMU corresponding in the visualized signal map from top to bottom is little finger, ring finger, middle finger, index finger, thumb, back of the hand, and wrist, the middle line of subfigures showing the signals of three channels of accelerometer and three channels of gyroscope for the thumb, and the bottom line of subfigures showing the corresponding gesture illustrations as well as the gesture numbers.

Table IX. Specifically, Yu et al. [9] combined force sensitive resistors (FSRs) and IMU to recognize 15 gestures using the back propagation (BP) neural network and bidirectional gated recurrent unit (Bi-GRU) network for UAV control. Zhang et al. [21] employed FSR, combining threshold with SVM and DTW to recognize eight gestures. Pan et al. [24] proposed a hierarchical framework based on sEMG and IMU data, and 21 gestures are recognized by the combination of SVM and DBN. In addition, to demonstrate the superiority of the hierarchical model, typical nonhierarchical models such as Zhang et al. [19], Maragliulo et al. [27], and Duan et al. [28] are selected for comparison as well. Zhang et al. [19] extracted features from IMUs, EMG, and pressure data to recognize gestures using LSTM. Maragliulo et al. [27] used sEMG data to extract features and utilized SVM to classify five gestures. Duan et al. [28] utilized LDA to recognize gestures and obtained the features of sEMG data through root mean square ratio (RMSR) and AR models. Compared with [9], [19], and [24], this work only relies on IMUs, which simplifies the configuration of sensors. Moreover, in contrast to [21], [27], and [28], more gestures are defined and can be recognized accurately.

In terms of the performance of gesture recognition, all the accuracy of results is derived based on the dataset of this work. As for the static gestures, the method in [21] performed the best. However, in the dynamic gesture dataset and all gesture dataset, except for the methods of Zhang et al. [21] and Maragliulo et al. [27], the accuracy of other methods under 10-foldCV and LOPOCV exceeded 80%. For a hierarchical model of Zhang et al. [21], since DTW matches sequences based on the gesture template library and is relatively good at recognizing known gestures, the method performed poorly in terms of the recognition accuracy of dynamic gestures. Regarding the real-time performance, the method of Zhang et al. [21] has the highest latency compared to the others.

In addition, the proposed two-stage model significantly outperformed other models, which indicated that the deep neural network can recognize dynamic gestures with complicated signal properties more accurately with less time loss. Meanwhile, the proposed method performs the best under LOPOCV, indicating that it can significantly enhance the model robustness under cross-subject recognition. This can be attributed to the focus on the signal properties of gestures, reducing the volume of data by differentiating between static and dynamic gestures. To a certain extent, it ensures that the data distribution is more uniform to eliminate the interference of wrong gestures. Overall, our model combines the advantages of a hierarchical framework and deep neural network to explore the equilibrium between accuracy and latency.

IV. Discussion and Limitation

Fig. 11 shows the confusion matrices when recognizing the 15 gestures based on four models, i.e., the two-stage model, nontwo-stage model, Zhang's work [21], and Pan's work [24]. As shown in Fig. 11(b)–(d), G8 (static), G9 (static), G14 (dynamic), and G15 (dynamic) are more frequently misclassified among each other, and related gesture signals are visualized in Fig. 12. We observed from Fig. 12(b), (e), and (f) that the signals (especially accelerometer signals) of the static

gesture G8 are quite similar to those of the dynamic gestures G14 and G15, except for certain dynamic details when the palm is moving horizontally based on the fingers being fixed. The similarity leads to the occurrence of misclassification. In addition, movements of the dynamic gestures G9, G14, and G15 lead to relatively small gyroscope signal fluctuations compared with the angular velocities of the static gesture G8. Since the nontwo-stage model does not specifically consider the properties of static and dynamic gestures, misclassification between static and dynamic gestures occurs. Although the methods of Zhang et al. [21] and Pan et al. [24] consider the properties of gestures, the generalization ability of the classifier used in the second stage is limited. Interestingly, the proposed two-stage model, as shown in Fig. 11(a), avoids the misclassification among the static gesture G8 and the dynamic gestures G14 and G15. This can be attributable to the consideration of respective gesture properties and stronger generalization ability based on deep neural network.

Nevertheless, gesture G3 is likely to be confused with G8 as illustrated in Fig. 11(a), which is due to the subtle differences at the thumb [middle subfigures of Fig. 12(a) and (b)] and the misclassification of the second stage. Besides, the gesture signals of G13 is distinguishable as visualized in Fig. 12(c), and it has been accurately recognized by all the four models. Therefore, it is necessary to further optimize the feature selection and consider the threshold-based differentiation in the first stage to improve the performance of gesture recognition. In addition, designing a set of discernible gestures with pleasant interaction experience is essential for constructing an HCI system.

V. Conclusion

We propose a two-stage framework that can accurately recognize gestures made by subjects wearing the self-developed data gloves for UAV control. The framework seeks a balance between traditional machine learning methods and deep neural networks. In order to acquire the optimal set of features, the effect of FCBF is investigated in terms of model accuracy and speed. In the proposed method, the differentiation between the properties of static and dynamic gestures is introduced. In the first stage, the hyperparameters of each classifier are optimized, and static gestures are distinguished from dynamic gestures. In the second stage, the classifier is used to classify static gestures, and the deep neural network is applied to classify dynamic gestures. The experimental results show that our method achieves an accuracy of 98.27% under LOPOCV, which outperforms the other six related works, with significant advantages on dynamic gestures recognition. However, IMU data are susceptible to the body movement, which will be further studied in the future. In addition, more comprehensive feature selection considering the properties of IMUs will be conducted, and the threshold-based differentiation will be combined with machine learning method to improve the accuracy of static gesture classification.

References

[1] L. Guo, Z. Lu, and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Trans. Hum.-Mach. Syst.*, vol. 51, no. 4, pp. 300–309, Aug. 2021.

[2] X. Song, S. S. Van De Ven, L. Liu, F. J. Wouda, H. Wang, and P. B. Shull, "Activities of daily living-based rehabilitation system for arm and hand motor function retraining after stroke," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 621–631, 2022.

[3] E. Mencarini, A. Rapp, L. Tirabeni, and M. Zancanaro, "Designing wearable systems for sports: A review of trends and opportunities in human–computer interaction," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 4, pp. 314–325, Aug. 2019.

[4] X. Zhang, Y. Sun, and Y. Zhang, "Evolutionary game and collaboration mechanism of human-computer interaction for future intelligent aircraft cockpit based on system dynamics," *IEEE Trans. Human-Mach. Syst.*, vol. 52, no. 1, pp. 87–98, Feb. 2022.

[5] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst. Man Cybern., C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007.

[6] S. Poularakis and I. Katsavounidis, "Low-complexity hand gesture recognition system for continuous streams of digits and letters," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2094–2108, Sep. 2016.

[7] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.

[8] M. Aggravi, C. Pacchierotti, and P. R. Giordano, "Connectivity-maintenance teleoperation of a UAV fleet with wearable haptic feedback," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1243–1262, Jul. 2021.

[9] C. Yu, S. Fan, Y. Liu, and Y. Shu, "End-side gesture recognition method for UAV control," *IEEE Sensors J.*, vol. 22, no. 24, pp. 24526–24540, Dec. 2022.

[10] C. Lu et al., "Online hand gesture detection and recognition for UAV motion planning," *Machines*, vol. 11, no. 2, p. 210, Feb. 2023.

[11] Z. Gao, Y. Wang, X. Sun, P. Chen, and C. Ma, "A multifeatured time–frequency neural network system for classifying sEMG," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 11, pp. 4588–4592, Nov. 2022.

[12] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Sep. 2019, pp. 10965–10974.

[13] M. Li, J. Wang, and N. Sang, "Latent distribution-based 3D hand pose estimation from monocular RGB images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4883–4894, Dec. 2021.

[14] X. Liang, D. Zhang, G. Lu, Z. Guo, and N. Luo, "A novel multicamera system for high-speed touchless palm recognition," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 51, no. 3, pp. 1534–1548, Mar. 2021.

[15] R. Zhao, K. Wang, R. Divekar, R. Rouhani, H. Su, and Q. Ji, "An immersive system with multi-modal human-computer interaction," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Xian, China, May 2018, pp. 517–524.

[16] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with Kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[17] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016.

[18] B. Tekin, F. Bogo, and M. Pollefeys, "H+O: Unified egocentric recognition of 3D hand-object poses and interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4511–4520.

[19] X. Zhang, Z. Yang, T. Chen, D. Chen, and M. Huang, "Cooperative sensing and wearable computing for sequential hand gesture recognition," *IEEE Sensors J.*, vol. 19, no. 14, pp. 5775–5783, Jul. 2019.

[20] P. B. Shull, S. Jiang, Y. Zhu, and X. Zhu, "Hand gesture recognition and finger angle estimation via wrist-worn modified barometric pressure sensing," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 724–732, Apr. 2019.

[21] Y. Zhang, B. Liu, and Z. Liu, "Recognizing hand gestures with pressure-sensor-based motion sensing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1425–1436, Dec. 2019.

[22] P. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 485–494, Apr. 2015.

[23] B. G. Lee and S. M. Lee, "Smart wearable hand device for sign language interpretation system with sensors fusion," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1224–1232, Feb. 2018.

[24] T. Pan, W. Tsai, C. Chang, C. Yeh, and M. Hu, "A hierarchical hand gesture recognition framework for sports referee training-based EMG and accelerometer sensors," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3172–3183, May 2022.

[25] S. Benatti et al., "A versatile embedded platform for EMG acquisition and gesture recognition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 5, pp. 620–630, Oct. 2015.
[26] W. Dong, L. Yang, R. Gravina, and G. Fortino, "Soft Wrist–Worn multifunctional sensor array for real-time hand gesture recognition," *IEEE Sensors J.*, vol. 22, no. 18, pp. 17505–17514, Sep. 2022.
[27] S. Maragliulo, P. F. A. Lopes, L. B. Osório, A. T. De Almeida, and M. Tavakoli, "Foot gesture recognition through dual channel wearable EMG system," *IEEE Sensors J.*, vol. 19, no. 22, pp. 10187–10197, Nov. 2019.
[28] F. Duan, X. Ren, and Y. Yang, "A gesture recognition system based on time domain features and linear discriminant analysis," *IEEE Trans. Cogn. Develop. Syst.*, vol. 13, no. 1, pp. 200–208, Mar. 2021.
[29] W. Chen, L. Feng, J. Lu, and B. Wu, "An extended spatial transformer convolutional neural network for gesture recognition and self-calibration based on sparse sEMG electrodes," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 6, pp. 1204–1215, Dec. 2022.
[30] Y.-T. Liu, Y.-A. Zhang, and M. Zeng, "Novel algorithm for hand gesture recognition utilizing a Wrist–Worn inertial sensor," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10085–10095, Dec. 2018.
[31] J. Galka, M. Masior, M. Zaborski, and K. Barczewska, "Inertial motion sensing glove for sign language gesture acquisition and recognition," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6310–6316, Aug. 2016.
[32] D. Zhang et al., "Fine-grained and real-time gesture recognition by using IMU sensors," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2177–2189, Apr. 2023.
[33] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *IEEE Sensors J.*, vol. 15, no. 1, pp. 154–163, Jan. 2015.
[34] Z. Wang et al., "Hear sign language: A real-time end-to-end sign language recognition system," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2398–2410, Jul. 2022.
[35] Y. Wu, Z. Wu, and C. Fu, "Continuous arm gesture recognition based on natural features and logistic regression," *IEEE Sensors J.*, vol. 18, no. 19, pp. 8143–8153, Oct. 2018.
[36] K. Czuszynski, J. Ruminski, and A. Kwasniewska, "Gesture recognition with the linear optical sensor and recurrent neural networks," *IEEE Sensors J.*, vol. 18, no. 13, pp. 5429–5438, Jul. 2018.
[37] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 5, pp. 1281–1290, Sep. 2016.
[38] A. Calado, V. Errico, and G. Saggio, "Toward the minimum number of wearables to recognize signer-independent Italian sign language with machine-learning algorithms," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.
[39] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

Buyuan Zhang received the B.S. degree from the Department of Computer Science, Beijing Institute of Technology, Beijing, China, in 2018. He is currently pursuing the M.S. degree with the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing.

His research interests include machine learning, signal processing, and human–machine interaction.

Haoyang Zhang received the Ph.D. degrees from the School of Precision Instrument and Opto-Electronics Engineering, Tianjin University, Tianjin, China, and the School of Mechanical and Materials Engineering, University College Dublin, Dublin, Ireland, in 2021.

He is currently a Research Assistant Professor with the Defense Innovation Institute, Academy of Military Sciences, Beijing, China. His main research interests include human–computer interaction and micro-/nanomanufacturing.

Tao Zhen received the Ph.D. degree from the College of Engineering, Beijing Forestry University, Beijing, China, in 2023.

He is currently a Research Assistant with the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing. His current research interests include gesture recognition and exoskeleton robotics.

Bowen Ji (Member, IEEE) received the Ph.D. degree in electronic science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2019.

From December 2016 to May 2018, he was a joint Ph.D. Student with the Prof. Yonggang Huang's Group, Northwestern University, Chicago, IL, USA. He is currently an Associate Professor with the Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, China. His research interests include implantable sensors as brain–computer interface and wearable flexible sensors for human–computer interaction.

Liang Xie received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2012, 2014, and 2018, respectively.

He is currently an Associate Researcher with the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China. His research interests include computer vision, human–machine interaction, and mixed reality.

Ye Yan received the B.S. and M.S. degrees from the Department of Automatic Control, National University of Defense Technology, Changsha, China, in 1994 and 1997, respectively, and the Ph.D. degree in aircraft design from the National University of Defense Technology, in 2000.

He worked as a Lecturer, an Associate Professor, and a Professor with the National University of Defense Technology. He is currently a Professor with the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China. His research interests include human–machine interaction and mixed reality.

Erwei Yin received the M.S. and Ph.D. degrees from the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, China, in 2010 and 2015, respectively.

He is currently a Researcher with the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China. His research interests include brain–computer interfaces and intelligent human–machine interaction technologies.