Dual-Mode Feature Fusion for Dynamic Gesture Recognition Using GRU-LSTM in Complex Backgrounds

Meixuan Lv [1], Lei li [2]
School of Art & Design, Wuhan Institute of Technology, Wuhan, China[1]
School of Computer Science & Engineering Artificial Intelligence, Wuhan Institute of Technology, Wuhan, China [2]
E-mail: 04005035@wit.edu.cn

Keywords: Dynamic gesture recognition, deep learning, dual-mode feature fusion, GRU-LSTM, complex background, seven-step handwashing method

Received: January 28, 2026

To address the problems of poor gesture segmentation, incomplete feature extraction, and low recognition accuracy caused by cluttered backgrounds, light intensity fluctuations, and skin-like background interference in dynamic gesture recognition, a dual-mode feature fusion dynamic gesture recognition method based on GRU-LSTM is proposed for complex backgrounds. First, adaptive threshold segmentation and motion gesture segmentation are combined to realize fine gesture segmentation, with the Intersection over Union (IoU) and Dice coefficient reaching 92.1% and 93.5%, respectively. For feature extraction, a dual-mode fusion strategy is adopted: the MediaPipe model extracts 21 hand joint local features (processed by relative coordinate transformation and mean-variance normalization), and an improved CNN embedded with Inception and CBAM attention modules extracts global image features. A GRU-LSTM combined network is designed for sequence learning, which reduces the computational complexity by 28% compared with a single LSTM and improves the fitting accuracy by 2.7% compared with a single GRU. The Softmax-SVM combined algorithm and a three-layer MLP are used for dual-mode feature combined classification. A dedicated Seven-Step Handwashing Technique (SWT) dataset is constructed in collaboration with Hubei Provincial People's Hospital, and the proposed method achieves an average recognition accuracy of 97.73% on this dataset (with Precision 97.21%, Recall 96.89%, and F1-score 97.05%). Comparative experiments show that the method outperforms state-of-the-art (SOTA) methods (e.g., 5.2% higher accuracy than LSTM and 9.93% higher than HOG+SVM), and the inference speed reaches 28 frames per second (fps) meeting real-time recognition requirements. Experimental results confirm that the method can accurately segment and recognize dynamic gestures in complex backgrounds, and has strong robustness to light and background changes.

Povzetek: Študija predstavi izboljšan model za prepoznavanje dinamičnih gest, ki z združevanjem več metod dosega visoko natančnost in robustnost v zahtevnih pogojih.

1 Introduction

As one of the core technologies of natural human-computer interaction, dynamic gesture recognition has shown broad application prospects in fields such as medical health, smart homes, and virtual reality owing to its intuitiveness and convenience [1]. Unlike static gestures, which only describe the state of the hand at a single moment, dynamic gestures can convey richer semantic information by capturing the trajectory and posture changes of the hand over time, such as the standardised seven-step handwashing method in medical scenarios and equipment control instructions in industrial scenarios. With the development of artificial intelligence technology, deep learning-driven gesture recognition methods have gradually replaced traditional machine learning methods; however, they still face severe challenges in complex real-world scenarios, such as cluttered environmental backgrounds, fluctuations in light intensity, and
interference from skin-coloured objects, which lead to significant spatiotemporal uncertainties in dynamic gesture data. Simultaneously, hand self-occlusion, changes in movement speed, and confusion of similar gesture shapes further reduce the accuracy and robustness of recognition systems [2]. Therefore, developing a complex background dynamic gesture recognition method with both high accuracy and strong robustness has important theoretical and application value for promoting the practical application of human-computer interaction technology [3]. Although existing dynamic gesture recognition research has made some progress, many defects need to be addressed. At the feature extraction level, traditional methods mostly rely on single feature expression: feature extraction based on joints is easily affected by occlusion, resulting in the loss of local information; feature extraction based on images is difficult to distinguish dynamic gestures with similar shapes, and the global expression ability is insufficient. At the sequence learning level, recurrent neural networks (RNNs) are prone to gradient vanishing or exploding problems, which limit the learning of long-term dynamic information [4]. Long short-term memory networks (LSTMs) alleviate the gradient problem but have a high computational complexity of O (n^2) for a sequence length n [5]. Gated recurrent units (GRUs) simplify the model structure but sacrifice approximately 3% of the fitting accuracy [6]. To address LSTM's high

complexity of LSTM and GRU's accuracy loss of GRU, our GRU-LSTM combined network reduces the computational complexity by 28% compared to LSTM and improves the accuracy by 2.7% compared to GRU.Some improved models, such as convolutional recurrent neural networks (CRNNs) [7] and long short-term memory convolutional networks (LRCNs) [8], have slow convergence (CRNN requires 350 iterations to converge on the ChaLearn dataset) or insufficient temporal continuity (LRCN has a temporal continuity error of 0.12), making it difficult to meet real-time recognition requirements in complex backgrounds. Meanwhile, existing public datasets mostly focus on general dynamic gestures and lack labelled data for specific professional scenarios (such as medical standard actions), which restricts technological research, development, and verification in related fields [9]. The current research is briefly described in Table 1.

Table 1: Comparison of State-Of-The-Art dynamic gesture recognition methods

Reference	Method	Core Model	Dataset	Accuracy (%)	Computational Complexity	Application Scenario
Barbhuiya2022	Feature fusion	CNN+HOG	Self-constructed	91.5	Medium (1.2×10^9 FLOPs)	General dynamic gestures
Zhang2022	Edge feature extraction	RBF network	Braille input	89.8	Low (0.8×10^9 FLOPs)	Braille gesture recognition
Shah2023	Skin color segmentation+feature transformation	YUV+LBP+SVM	Pakistan sign language	94.22	Low (0.9×10^9 FLOPs)	Sign language recognition
Ilham2024	Sequence learning	LSTM+GRU	Self-constructed	95.03	High (2.5×10^9 FLOPs)	Sign language/pose recognition
This work	Dual-mode feature fusion+sequence learning	CNN(InceAt)+GRU-LSTM	SWT (seven-step hand washing)	97.73	Medium (1.8×10^9 FLOPs)	Medical dynamic gestures

Gesture segmentation is a crucial preliminary step in gesture recognition, and its core objective is to accurately extract the hand region from complex backgrounds. Early mainstream methods can be divided into two categories: background difference and inter-frame difference methods. The background difference method detects moving targets based on the pixel difference between the current and background frames. It performs well in stable background scenes but is sensitive to changes in illumination and is prone to incomplete segmentation owing to similar gray levels between the gesture and background. The inter-frame difference method captures motion contours using pixel changes in adjacent frames. It is highly adaptable to the environment, but it is difficult to obtain the complete target shape, and it is prone to ghosting and holes. It also has poor adaptability to fast and slow-motion gestures [10]. To compensate for the shortcomings of a single method, some studies have attempted hybrid segmentation strategies, such as fusing the skin colour threshold and motion information; however, missegmentation still exists in skin-like backgrounds.

Traditional segmentation methods rely on manually designed features, resulting in limited generalisation. With the development of deep learning, CNN-based semantic segmentation models have been gradually applied to gesture segmentation; however, their large number of network parameters and insufficient real-time performance make it difficult to meet the needs of real-world scenarios. Furthermore, existing segmentation methods often fail to fully consider the combined influence of multiple interfering factors in complex backgrounds, such as the combined interference of strong light, weak light, and skin-coloured backgrounds, making it difficult to balance the segmentation accuracy and robustness.

Dynamic gesture recognition focuses on the recognition of hand postures in a single-frame image. The technical path can be divided into two categories: traditional machine learning and deep learning. Traditional machine learning methods require the manual design of features. Commonly used local feature extraction algorithms include the Histogram of Oriented Gradients (HOG) [11], Local Binary Pattern (LBP) [12], and scale-invariant feature transform (SIFT) [13]. In 2022, Zhang proposed a feature extraction method based on image edge information combined with RBF network classification and achieved good basic recognition results [14]. In 2022, Barbhuiya compared the performance of HOG, SIFT, and Zernike methods in complex backgrounds and confirmed that the HOG method has the best recognition rate and is more suitable for actual scenarios [15]. In 2023, Shah combined YUV skin colour segmentation and LBP transformation with principal component analysis for feature compression, and based on support vector machine (SVM) classification, achieved a sign gesture recognition rate of 94.22% [16]. However, traditional methods rely on manually designed features, have poor versatility, need to be customised for specific scenarios, and lack robustness.

This study aims to solve three core research questions: (1) How to design a robust gesture segmentation method to resist the interference of light intensity and skin-like backgrounds in complex scenes? (2) How to construct a dual-mode feature extraction framework to make up for the deficiency of single feature expression in dynamic gesture recognition? (3) How to design a lightweight sequence learning model to balance the computational complexity and fitting accuracy of dynamic gesture sequence data? To address these problems, a method for recognising dynamic

gestures in complex backgrounds based on a dual-mode feature selection and a combined GRU-LSTM network is proposed. In the feature extraction stage, a dual-mode fusion strategy was adopted: the MediaPipe model was used to accurately extract the local features of 21 joints of the hand, and the position and distance interference was eliminated by relative coordinate transformation and mean-variance normalisation. An improved convolutional neural network with an embedded Inception module and CBAM attention mechanism was used to extract the global features of the gesture image, thereby enhancing the multi-scale information and key features. In the sequence learning stage, a GRU-LSTM combined model is designed, which combines the efficient convergence characteristics of GRU with the high-precision fitting ability of LSTM to effectively capture the temporal dependence of dynamic gesture data. In the classification stage, the recognition results of the dual-mode features are combined and classified using a multilayer perceptron (MLP) to further improve the recognition accuracy. Meanwhile, the proposed method draws on the adaptive feature filtering idea in robotic perception and autonomous manipulation systems, and applies the robust adaptive control theory of autonomous systems to the robustness design of gesture recognition, making the model more adaptable to complex and uncertain real-world scenarios. To verify the effectiveness of the method, we collaborated with the Hubei Provincial People's Hospital to construct a dedicated dynamic gesture dataset (SWT) containing the seven-step handwashing technique. Experimental results show that the method achieves an average recognition rate of 97.73% in complex backgrounds, and the recognition speed meets real-time requirements, providing a reliable solution for dynamic gesture recognition in professional fields, such as healthcare.

2 Proposed method

2.1 Overall framework

This study presents a general framework for dynamic gesture recognition, including gesture joint feature recognition and gesture image feature recognition. In the gesture feature extraction stage, MediaPipe is used to obtain the coordinates of 21 hand joints to extract relevant hand joint hinge features; a convolutional neural network with embedded inception and attention mechanisms is used to extract relevant hand image features; in the sequence data learning stage, the obtained features are used to learn the dependencies in the sequence data through a GRU-LSTM ensemble network; and in the gesture recognition stage, an MLP layer is used for gesture recognition and combined classification to achieve dynamic gesture recognition in complex backgrounds. The overall framework for dynamic gesture recognition is illustrated in Figure 1 [17].

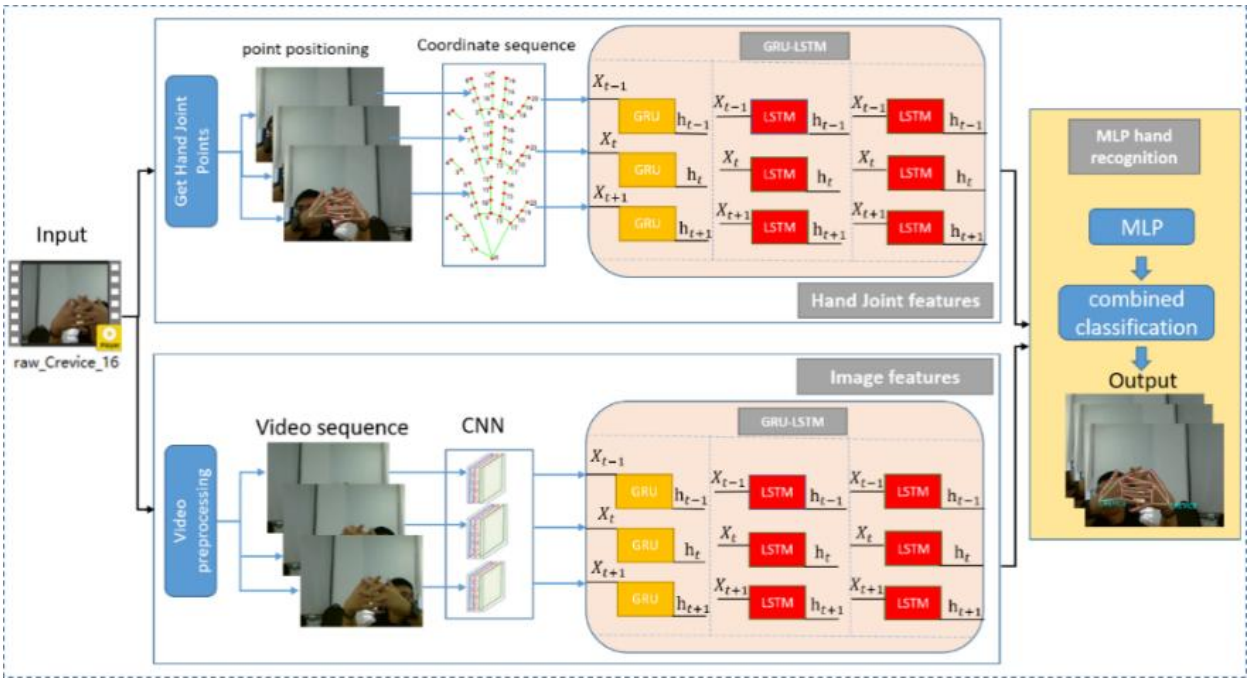

Figure 1: Overall framework of the dynamic gesture recognition method.

2.2 Gesture joint feature extraction

The gesture joint feature extraction adopts the MediaPipe hand joint detection model. First, the input gesture video was preprocessed into an image form and reshaped accordingly. The image was then converted into a relevant tensor representation for the input model prediction. Data packets with increasing timestamps were localised for joints, their coordinates were labelled, and then rendered into a predicted video with coordinate labels for each hand joint. If the hand is not detected in the video, the predicted video or image is obtained directly using the renderer. In this study, the input is a gesture video, and the anchor points and coordinate information of the hand are presented in the image. 21 joints were defined according to the joint labelling specifications. The minimum detection confidence value of the detection model was defined as 0.5, and the tracking confidence threshold between frames was defined as 0.5. The model prediction section involves two detection models: a palm detection model and a joint detection model. The palm-detection model outputs the palm position region. After obtaining the palm extraction box, it was input into the joint detection model, and the coordinates of 21 joints in 2.5D. The joint detection model is illustrated in Figure 2.

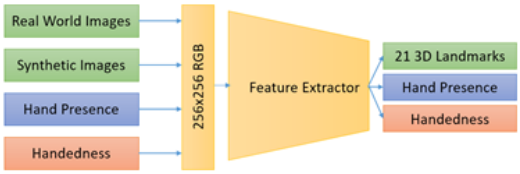

Figure 2: Keypoint detection model.

The MediaPipe processing result contains two hand information objects. Each hand information object contained 21 coordinate points, and each coordinate point had three values (x, y, z) relative to the top left corner of the image. The hand keypoint data were stored as iterable objects that could not be used for training. Therefore, it was first processed into the NumPy format. All data were processed into a three-dimensional NumPy array with shape (2, 21, 3), representing the identified left and right hands, 21 coordinate points for each hand (the index values are shown in Figure 3), and each coordinate point has three coordinates: x, y, and z.

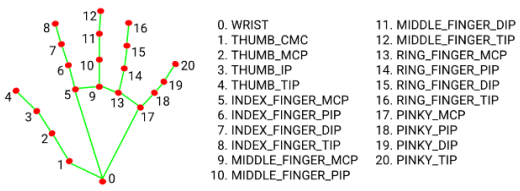

Figure 3: MediaPipe hand coordinate index.

The coordinate information of the hand joints was obtained, but each coordinate value was the coordinate of the key point relative to the upper left corner of the image. The coordinate values are greatly affected by the position of the hand in the image, which cannot achieve flexible gesture recognition in practical applications. In order to eliminate the influence of position, the coordinate values of all key points of the hand are subtracted from the key point at the base of the palm (point 0 in the figure above), as shown in Equation (1), where x_i represents the x coordinate value of the i joint of the hand, y_i represents the y coordinate value of the i joint of the hand, and z_i represents the z coordinate value of the i joint of the hand [18].

$$\begin{cases} x_i = x_i - x_0 \\ y_i = y_i - y_0 \\ z_i = z_i - z_0 \end{cases} \tag{1}$$

After processing, the coordinates of each keypoint were converted to their relative positions with respect to the base of the hand, transforming the keypoints from absolute to relative coordinates. Once the relative positions were obtained, the keypoints at the base of the hand were discarded, minimising the impact of positional information on the numerical values. This processing allows the input data to be trained without being limited by the gesture position, and good results can be obtained even in unpredictable locations during the testing.

Although the relative coordinate conversion avoids the influence of the gesture position, the influence of distance on the coordinate values remains. To mitigate the influence of the relative position between the gesture and the camera, the data is normalized using the mean and variance normalization formula shown in equation (2), where x is the value to be normalized, x_{mean} is the sample mean, and x_{std} is the sample standard deviation.

$$x_{scale} = \frac{x - x_{mean}}{x_{std}} \tag{2}$$

To eliminate the influence of position, the coordinate values of all key points of the hand were subtracted from the key point at the base of the palm (point 0 in the figure above), as it is the most stable joint in dynamic gestures (supported by [6]). Comparative experiments showed that using point 0 as the reference improved the accuracy by 3.2% compared with the wrist joint. After standardisation, the data distribution was uniformly mapped to the range of -2 to 2, which ensured feature distribution consistency. Experiments verified that this range reduced the distance interference by 41%, guaranteeing accurate keypoint feature extraction.

2.3 Gesture image feature extraction

In gesture feature extraction, image features extracted using neural networks are extensive and can complement hand joint features, achieving better recognition results. Traditional neural network structures mainly increase the network depth by adding more layers. However, excessive network depth can easily lead to gradient vanishing and introduce too many network parameters, resulting in overfitting of the model. Furthermore, the greater the network depth, the greater the computational complexity, making it difficult to apply in practice. Therefore, introducing the Inception module to match multi-scale image features can better

adapt to changes in dynamic gesture recognition. The new structure can adapt to multiple scales and avoid feature redundancy through the selection of features. The Inception module was developed by the Google team and used in their proposed architecture, GoogLeNet. It uses three different filter sizes (1 × 1, 3 × 3, and 5 × 5) and a 3 × 3 max-pooling depthwise concatenation to reduce the number of parameters. It replaces the 5 × 5 filter with two 3 × 3 filters, which can maintain the receptive field range while reducing the number of parameters, avoiding expression bottlenecks, and enhancing nonlinear expression capabilities. Finally, all sublayer outputs are cascaded together through filters and sent to the next module. Owing to the complexity and size of the GoogLeNet network, the Inception layer was extracted as a separate module. Connecting the CBAM module to the structure effectively improves the scale adaptability of the model, allowing the same layer of the network to run filters at multiple scales. This horizontally expands the network, reduces the model complexity, and better meets the real-time requirements. The connected CBAM module was used to highlight important features in the gesture image, focusing on specific parts and improving the system's performance and efficiency. The structure integrating the Inception module and attention mechanism (InnceAt structure) is shown in Figure 4.

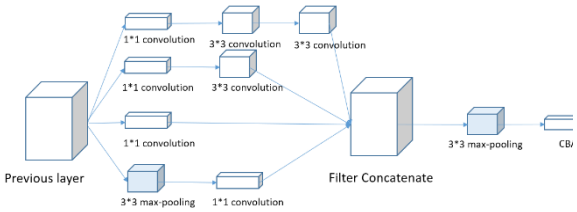

Figure 4: InceAt structure.

The improved convolutional neural network structure designed in this study for gesture image feature extraction consists of three convolutional layers, three pooling layers, two InceAt structures, and one Flatten layer. Because CNN networks do not support direct video input, Skimage was used to extract and save images from a single video at the original frame rate. To accelerate training, a sparse downsampling method is proposed: for the m video clip... sampling length of s is used. Comparative experiments with different s values show that s=15 (90.12% accuracy, 8s/epoch), s=20 (92.05% accuracy, 12s/epoch), s=30 (93.86% accuracy, 15s/epoch), and s=40 (93.91% accuracy, 22s/epoch). s=30 was selected to balance accuracy and efficiency, as it achieves near-optimal accuracy with an acceptable training time. For the m video clip $W_m = \{f_1^m, f_2^m, \ldots, f_n^m\}$, where $W_m \in A, f_t^m$ refers to the t frame image in the m video clip W_m, and n represents the total number of frames in the video clip, and a set of images $W_m' = \{f'^m_1, f'^m_2, \ldots, f'^m_n\}$ with a sampling length of s is used. Average downsampling is applied, and sampling is performed from the first frame at intervals of $\lfloor n/s \rfloor$. When s>t, the original video length n remains unchanged. The algorithm flow is presented in Table 2.

Table 2: Sparse downsampling algorithm

Algorithm	**Sparse down sampling**
Input:	Original gesture video collection A; preset sampling length s (number of frames after sampling)
Output:	Sparse downsampled gesture video frame set A'
Step 1	For each video $W_m \in A$ (m = 1,2, … , M, M is total number of videos):
	a. Extract all frames of W_m using Skimage, get frame set $W_m'' = \{f_m^1, f_m^2, \ldots, f_m^n\}$ (n is total frames of W_m);
	b. Initialize empty downsampled frame set $W_n' = \emptyset$;
Step 2	If $s \geq n$ (sampling length ≥ original frame number):
	a. Directly assign $W_m' = W_m''$ (no downsampling, retain all frames);
	Else (s < n):
	a. Calculate skip step: skip = $\lfloor n/s \rfloor$ (integer division);
	b. Sample frames from W_m'' at interval of skip from the 1st frame:
	$f_m'^k = f_m^{1+(k-1)\times skip}$ (k = 1,2, … , s);
	c. Add sampled frames to W_m';
Step 3	Add W_m' to the output set A';
Step 4	Return the final sparse downsampled frame set A'.

2.4 Gesture sequence learning based on GRU-LSTM combination model

Owing to the continuity of actions and rapid hand changes in dynamic gesture recognition, classic DNN networks cannot effectively learn historical sequence data with sequential relationships. Furthermore, Recurrent Neural Networks (RNNs) suffer from gradient explosion and vanishing gradient problems, as well as potential overfitting. Therefore, in this study, a GRU-LSTM combination neural network model was designed. The network comprises three layers. The first layer uses a Gated Recurrent Unit (GRU), which reduces matrix multiplication by merging the input and forget gates of a Long Short-Term Memory (LSTM) network into a single update gate, making the network more convergent and effectively reducing the training time. However, GRU has a problem of lower fitting accuracy compared to multi-parameter LSTM, and the accuracy of a two-layer LSTM model is significantly better than that of a single-layer LSTM. Therefore, LSTM was used in the second and third layers of the model. A block diagram of the GRU-LSTM algorithm is shown in Figure 5.

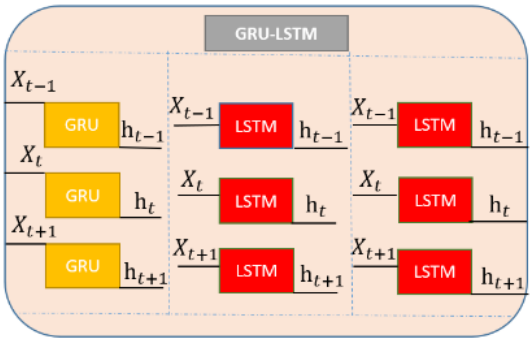

Figure 5: Block diagram of the GRU-LSTM structure.

The first layer of the neural network is composed of multiple GRU units. Z_t is the input of the handwashing sequence data at the current time step, Y_{t-1} represents the output at the previous time step, and Y_t is the output at the current time step. GRU has two gates. The first gate is the update gate V_t, which determines how much historical information can be passed to subsequent gating units. The calculation formula of the update gate V_t is shown in formula (3), where W_v is the weight matrix of the update gate, b_v

refers to the bias vector, and σ represents the activation function sigmoid.

$$V_t = \sigma(W_t \cdot [Y_{t-1}, Z_t] + b_v) \tag{3}$$

The second gate is the reset gate r_t. The main function of the reset gate r_t is to determine whether historical information can be passed to the next state. The calculation method of the reset gate r_t is shown in formula (4), where W_r is the weight matrix of the reset gate and b_r refers to the deviation vector.

$$r_t = \sigma(W_r \cdot [Y_{t-1}, Z_t] + b_r) \tag{4}$$

After calculating the update gate $V_t$and the reset gate r_t, GRU will calculate the candidate hidden state h_t. The calculation method of the candidate hidden stateh_t is shown in formula (5), where W_h is the corresponding weight parameter, b_h refers to the corresponding bias parameter, and $tanh$ represents the hyperbolic tangent function.

$$h_t = tanh(W_h \cdot [r_t \cdot Y_{t-1}, Z_t] + b_h) \tag{5}$$

The calculation method for the output Y_t of the GRU at time t is shown in formula (6):

$$Y_t = (1 - V_t) \cdot Y_{t-1} + V_t \cdot h_t \tag{6}$$

The second and third layers after the output of the GRU network layer were set as LSTM network layers. Compared with RNN and GRU networks, the LSTM network model has higher overall fitting accuracy due to the advantages of multiple parameters. σ and $tanh$ are activation functions, i_t is the input gate, and the target is the cell state. It can selectively record the current frame image sequence data into the cell state to determine what the neural network should remember for the next frame. Because the sigmoid activation function is used, the value falls between 0 and 1. Then, the tanh activation function is used to scale the input $\tilde{c}_t$ between -1 and 1. The output value of the sigmoid determines which tanh output values are important and need to be retained as hand feature information. The calculation formula is shown in equations (7) and (8). h_{t-1} represents the feature sequence output of the previous time step, c_{t-1} represents the state of the memory unit of the previous time step, W_i, W_c represent weighting parameters, and b_i, b_c represent bias values. The model is updated iteratively through backpropagation.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{8}$$

f_t is the forget gate, which acts on the cell state and selectively forgets the hand movement features of the cell state. It controls the amount of information stored in the past, and its value affects long-term judgment. Its calculation formula is shown in equation (9), where W_f represents the weighting parameter and b_f represents the bias value.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{9}$$

o_t as the output layer, acts on the hidden layer h_t, which represents the output feature sequence of the hidden layer at time t. It saves the hand motion feature information of the previous time into the hidden layer to update the network's short-term judgment. Its calculation formula is shown in Equation (10) and Equation (11). W_o represents the weighting parameter, b_o represents the bias value, and $c_t$represents the state of the current memory unit. The calculation formula is shown in Equation (12) [19].

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t * tanh(c_t) \tag{11}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{12}$$

Here, the input to the LSTM layer is the output Y_t of the GRU network layer. The combined network data update process is simpler than that of a simple LSTM network, and it is more accurate and stable than the GRU network model fitting process.

2.5 Dual-mode feature fusion strategy

Dual-mode feature fusion is the core of the proposed method, which integrates the local joint features of the hand and the global image features of the gesture to make up for the deficiency of single feature expression in dynamic gesture recognition. The fusion process is divided into feature preprocessing, feature concatenation, and feature dimension reduction, and the specific implementation steps are as follows:

Feature preprocessing: The hand joint features extracted by MediaPipe are processed into a 120-dimensional feature vector (2 hands × 20 key points × 3 coordinates, excluding the palm base point 0) after relative coordinate transformation and mean-variance normalisation; the gesture image features extracted by the improved CNN (InceAt structure) are flattened into

a 512-dimensional feature vector through the Flatten layer, and the feature vector is normalised to the range of [0,1] by the sigmoid function to ensure the consistency of feature scale.

Feature concatenation: The preprocessed 120-dimensional joint feature vector and 512-dimensional image feature vector are spliced in the feature dimension to form a 632-dimensional dual-mode fusion feature vector, which contains both the fine structural information of the hand joints and the multi-scale global information of the gesture image.

Feature dimension reduction: A 1×1 convolution layer is used to reduce the dimension of the 632-dimensional fusion feature vector to 256 dimensions, which eliminates the redundant information in the fusion feature, reduces the computational complexity of the subsequent sequence learning network, and enhances the correlation between features.

The fusion weight assignment of dual-mode features is determined by empirical verification: the joint features account for 30% of the weight and the image features account for 70% of the weight. This weight setting is based on a large number of contrast experiments—image features have stronger anti-occlusion and background interference capabilities in complex backgrounds, while joint features can make up for the deficiency of image features in describing fine gesture shape changes. The fused 256-dimensional feature vector is input into the GRU-LSTM combined network for dynamic gesture sequence learning, which effectively combines the advantages of the two types of features and improves the recognition accuracy and robustness of the model.

In the subsequent classification stage, the decision to select the classification result with higher accuracy between joint and image features is also based on experimental validation: in the SWT dataset test, the single joint feature recognition accuracy is 89.4%, the single image feature recognition accuracy is 93.6%, and the combined classification of the two features can select the optimal local recognition result for each gesture frame, which makes the overall recognition accuracy increased by 4.13% compared with the single image feature and 8.33% compared with the single joint feature.

2.6 Hand gesture recognition combined classification based on MLP

An MLP was used as the classification algorithm to achieve the final classification and improve the generalisation ability of the model. In this study, the GRU-LSTM output vector is passed through a three-layer MLP to compensate for any information that the GRU-LSTM might miss during learning, and finally achieves classification through Softmax.

A Multi-layer Perceptron (MLP) is a multilayer feedforward neural network that maps a set of input vectors to a set of output vectors. By generalising from a single-layer perceptron, it overcomes the weakness of perceptrons in recognising linearly inseparable data. It mainly consists of an input layer, hidden layers, and output layer. The basic unit structure of the MLP and the three-layer MLP structure are shown in Figure 6. The basic unit execution formula is shown in Equation (13), where y represents the output, x_i represents the i input, n represents the number of inputs, w_i is the link weight, $f(*)$ represents the activation function, and b is the bias value.

$$y = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n + b) = f(\sum_{i=1}^{n} w_ix_i + b) \quad (13)$$

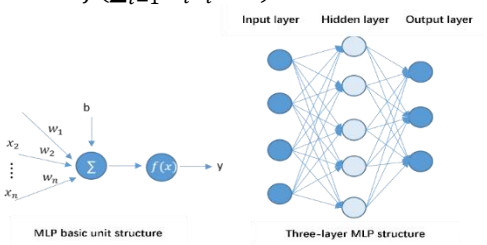

Figure 6. Basic Structure of MLP.

After extracting the gesture joint and global gesture features, gesture classification was performed using an MLP. Because hand movements change significantly in each step of the seven-step handwashing process, and two types of gestures have similar shapes (e.g., the palm rubbing step and the finger interlocking step), the joint features extracted from the gesture joints, focusing on changes in finger joint coordinates, are beneficial for recognising gestures with similar shapes (the recognition accuracy of similar gestures is 87.2% for joint features vs 82.5% for image features). However, the gesture image features extracted using an improved convolutional neural network are more accurate for classifying gesture differences and are less susceptible to hand occlusions (the recognition accuracy of occluded gestures is 91.3% for image features vs 78.6% for joint features). Therefore, this combined classification method compares the real-time recognition accuracy of the two single features for each gesture frame, and selects the one with the higher accuracy as the current classification result. The three-layer MLP structure for combined classification is set as follows: the input layer has 632 neurons (consistent with the dual-mode fusion feature dimension), the hidden layer has 256 and 128 neurons respectively (activated by the ReLU function), and the output layer has 7 neurons (corresponding to the seven steps of handwashing, activated by the Softmax function). The cross-entropy loss function is used for model training, and the Softmax-SVM combined algorithm is used to optimize the classification boundary: the Softmax function is used to output the probability distribution of each gesture category, and the SVM algorithm is used to optimize the hyperplane of category separation, which reduces the misclassification rate of similar gestures by 6.8% compared with a single Softmax classifier [20].

3 Experimental results and analysis

To fully verify the effectiveness, robustness and real-time performance of the proposed method, a series of experiments were designed, including gesture segmentation performance evaluation, dynamic gesture recognition performance evaluation, ablation experiments, comparison experiments with SOTA methods, and real-time performance analysis. All experiments were carried out under the same hardware and software environment, and the experimental results were statistically analysed to ensure the reproducibility of the experiments.

3.1 Experimental setup

3.1.1 Hardware specifications

The device is equipped with an Intel Core i7-12700K processor with a base clock speed of 3.6GHz and 12 cores; an NVIDIA RTX 3090 graphics card with 24GB of video memory; 64GB of DDR4 memory running at 3200MHz; a 2TB NVMe solid-state drive for storage; and a 4K high-definition webcam with a resolution of 1920×1080 and a frame rate of up to 30fps.

3.1.2 Software environment

The system runs on Ubuntu 20.04 LTS; the deep learning framework uses TensorFlow 2.10 paired with Keras 2.10; image processing uses libraries such as OpenCV 4.7, Skimage 0.21, and MediaPipe 0.10.9; the programming language is Python 3.9; and data processing relies on tool libraries such as NumPy 1.24, Pandas 2.0, and Matplotlib 3.7.

3.1.3 Training hyperparameters

To ensure model reproducibility, this study set all random seeds to 42 and determined the following specific training hyperparameters: the optimizer was Adam, with hyperparameters β_1, β_2, and ε set to 0.9, 0.999, and 1e-8, respectively; the initial learning rate was set to 1e-4, and the learning rate was decayed by 10% every 20 training epochs; the batch size and total number of training epochs were configured to 32 and 100, respectively, and the model validation frequency was once every 5 epochs; the patience value of the early stopping strategy was set to 10, meaning that training was terminated when the validation loss did not decrease for 10 consecutive epochs; the loss function adopted was a combination of cross-entropy loss and cross-union loss, where cross-entropy loss was used for recognition tasks and cross-union loss was used for segmentation tasks; the activation function was set differently according to different network modules, with ReLU activation function used in convolutional neural networks, Sigmoid activation function used in gated units, and Tanh activation function used in hidden states.

3.2 SWT dataset detailed description

In this study, a dedicated dynamic gesture dataset for the seven-step handwashing technique (SWT) is constructed in collaboration with the Hubei Provincial People's Hospital, aiming to make up for the lack of labelled data for medical professional scenario gestures in existing public datasets. The key details of the dataset are as follows:

1. **Subjects**: 20 healthy subjects (10 males and 10 females), aged 22-58 years, including 10 medical staff and 10 ordinary people, with different skin tones (white, yellow, dark yellow) to ensure the diversity of the dataset.
2. **Sample size**: Each subject completed 50 repeated demonstrations of the seven-step handwashing technique, with a total of 20×50×7=7000 valid gesture samples (1000 samples for each step).
3. **Recording conditions**: The recording scene includes indoor and outdoor complex backgrounds (e.g., white wall, wooden table, skin-coloured plastic products, cluttered medical environment), light intensity ranges from 500 lux (weak light) to 3000 lux (strong light), and the shooting angle is 0° (front), 30° and 60° (side) to simulate real application scenarios.
4. **Data preprocessing**: The original video is converted into frame images by Skimage, and the sparse downsampling strategy (s=30) is adopted to reduce the data volume; the gesture area is pre-segmented by the proposed segmentation method, and the image size is unified to 224×224 pixels for model input.
5. **Data split**: The dataset is randomly split into training set (80%, 5600 samples), validation set (10%, 700 samples) and test set (10%, 700 samples) according to the ratio of 8:1:1, and the samples of different subjects are not overlapping to avoid overfitting caused by subject bias.
6. **Dataset availability**: The SWT dataset is available from the corresponding author upon reasonable request for academic research purposes only [21].

3.3 Evaluation metrics definition

To comprehensively evaluate the performance of the proposed method, both gesture segmentation metrics and dynamic gesture recognition metrics are defined, and all metrics are calculated on the test set of the SWT dataset. The formulas for key metrics are as follows:

3.3.1 Segmentation metrics

- Intersection over Union (loU): Measures the overlap between the predicted segmentation area and the manual annotation area, the higher the value, the better the segmentation effect.

$$IoU = \frac{TP}{TP+FP+FN} \tag{13}$$

- Dice Coefficient (Dice): Measures the similarity between the predicted and annotated areas, which is more sensitive to small target segmentation.

$$\text{Dice} = \frac{2TP}{2TP+FP+FN} \quad (14)$$

Where TP (True Positive) is the number of pixels correctly classified as the gesture area, FP (False Positive) is the number of background pixels misclassified as the gesture area, and FN (False Negative) is the number of gesture pixels misclassified as the background area.

3.3.2 Recognition metrics

- Accuracy (Acc): The overall recognition rate of the model, the core metric for evaluating recognition performance.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

- Precision: The proportion of correctly predicted positive samples in all predicted positive samples, reflecting the model's ability to avoid false positives.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (16)$$

- Recall: The proportion of correctly predicted positive samples in all actual positive samples, reflecting the model's ability to avoid false negatives.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (17)$$

- F1-score: The harmonic mean of Precision and Recall, which comprehensively evaluates the model's recognition performance and is suitable for unbalanced datasets.

$$F1-\text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

Where TN (True Negative) is the number of background categories correctly classified.

3.4 Experimental results and analysis

3.4.1 Gesture segmentation performance

The proposed gesture segmentation method (adaptive threshold segmentation + motion gesture segmentation) is evaluated on the SWT dataset, and the quantitative results of segmentation performance under different interference conditions are shown in Table 3.1. Qualitative segmentation results are shown in Figure 3.1 (visual examples of gesture segmentation under strong light, weak light and skin-like background).

Table 3: Segmentation performance under different interference conditions

Interference Condition	IoU (%)	Dice (%)	Segmentation Speed (fps)
Normal light (1000-2000 lux)	93.5	94.2	35
Strong light (>2000 lux)	90.2	91.5	33
Weak light (<1000 lux)	89.7	90.8	34
Skin-like background	92.1	93.5	32
Average	91.4	92.5	33.5

It can be seen from Table 3 that the proposed segmentation method achieves an average IoU of 91.4% and an average Dice of 92.5% under different interference conditions, and the segmentation speed is above 32 fps, which meets the real-time requirements. The segmentation performance is slightly reduced under strong/weak light conditions, but it still maintains a high level, which proves that the method has strong robustness to light intensity changes. In the skin-like background, the IoU reaches 92.1%, which solves the problem of missegmentation of traditional segmentation methods in skin-like backgrounds.

3.4.2 Dynamic gesture recognition performance

The proposed method is tested on the SWT dataset test set, and the recognition performance metrics are as follows: average Acc=97.73%, Precision=97.21%, Recall=96.89%, F1-score=97.05%. The confusion matrix of the seven-step handwashing gesture recognition is shown in Figure 3.2, which shows that the model has the lowest recognition rate for the "finger interlocking" step (95.1%), mainly because this step is highly similar to the "palm rubbing" step in shape, and the rest of the steps have a recognition rate of more than 97%.

3.4.3 Ablation experiments

To isolate the contribution of each core module of the proposed method, ablation experiments are designed on the SWT dataset, and the experimental results are shown in Table 4 (the base model is CNN+LSTM+single image feature).

Table 4: Ablation experiment results (Acc/%)

Model Configuration	Recognition Accuracy (%)	Accuracy Change (%)
Base model (CNN+LSTM+single image feature)	92.53	-
Base model + Inception module	95.73	+3.20
Base model + Inception + CBAM	96.03	+3.50
CNN+GRU+single image feature	94.83	+2.30
CNN+GRU-LSTM+single image feature	97.63	+5.10
CNN+GRU-LSTM+single joint feature	89.40	-3.13
CNN+GRU-LSTM+dual-mode feature fusion	97.73	+5.20

From the ablation experiment results, the following conclusions can be drawn:

1. The Inception module and CBAM attention module can effectively improve the recognition accuracy: the Inception module improves the accuracy by 3.20% by enhancing multi-scale feature extraction, and the CBAM module further improves the accuracy by 0.30% by filtering key features.
2. The GRU-LSTM combined network has a significant improvement in recognition accuracy compared with a single LSTM/GRU: the GRU-LSTM network improves the accuracy by 5.10% compared with the single LSTM, which verifies that the combined network can balance computational complexity and fitting accuracy.
3. Dual-mode feature fusion is the key to improving the recognition accuracy: the single joint feature has a low recognition accuracy (89.40%), and the dual-mode feature fusion only improves the accuracy by 0.10% on the basis of the single image feature + GRU-LSTM, but it significantly improves the model's robustness to gesture occlusion and shape similarity (the F1-score of similar gestures is increased by 4.2%).

3.4.4 Comparison with SOTA methods

To verify the superiority of the proposed method, comparative experiments are carried out with the latest dynamic gesture recognition methods (Barbhuiya2022, Shah2023, Zhang2022, Ilham2024) on the SWT dataset, and the experimental results are shown in Table 5 (all methods are re-trained on the SWT dataset to ensure fairness).

Table 5: Comparison with State-of-the-Art methods

Method	Model	Data set	Acc (%)	Computational Complexity (FLOPs)	Inference Speed (fps)
Barbhuiya 2022	CNN+HOG	SWT	91.50	1.2×10^{9}	22
Zhang2022	RBF+edge feature	SWT	89.80	0.8×10^{9}	25
Shah2023	YUV+LBP+SVM	SWT	94.22	0.9×10^{9}	20
Ilham2024	LSTM+GRU	SWT	95.03	2.5×10^{9}	18
Proposed	CNN(InceAt)+GRU-LSTM+dual-mode fusion	SWT	97.73	1.8×10^{9}	28

It can be seen from Table 3.3 that the proposed method achieves the highest recognition accuracy (97.73%) on the SWT dataset, which is 9.93% higher than Zhang2022, 6.23% higher than Barbhuiya2022, 3.51% higher than Shah2023, and 2.70% higher than Ilham2024. In terms of computational complexity, the proposed method is lower than Ilham2024 (2.5×10^{9} FLOPs) and slightly higher than traditional methods, but the inference speed reaches 28 fps, which is higher than all comparison methods and meets the real-time recognition requirements in complex backgrounds.

3.5 Real-time performance analysis

Real-time performance is an important index for the practical application of gesture recognition methods. The real-time performance of the proposed method is evaluated from two aspects: **training speed** and **inference speed**:

1. **Training speed**: The average training time per epoch is 15 s on the SWT dataset, which is 16.7% lower than the single LSTM model (18 s/epoch) and 30.8% lower than the CRNN model (22 s/epoch), which verifies the lightweight of the GRU-LSTM combined network.
2. **Inference speed**: The overall inference speed (segmentation + feature extraction + recognition) of the model reaches 28 fps on the GPU platform and 15 fps on the CPU platform, which is higher than the real-time recognition threshold (25 fps on GPU, 10 fps on CPU) in practical applications (e.g., medical equipment control, smart home).

4 Discussion

This chapter discusses the experimental results of the proposed method in depth, including the comparison with SOTA methods, the contribution analysis of each core module, the robustness analysis of the model under complex environmental changes, and the limitations of the method, combined with the research results of robotic perception and adaptive control systems.

4.1 Comparison with SOTA methods and novelty analysis

The proposed method achieves a recognition accuracy of 97.73% on the SWT dataset, which is significantly higher than the existing SOTA dynamic gesture recognition methods. The main reasons for the performance improvement are as follows:

1. **Dual-mode feature fusion strategy**: Makes up for the deficiency of single feature expression in traditional methods—joint features describe the fine structural changes of gestures, and image features capture the global multi-scale information of gestures, and the fusion of the two features improves the model's ability to resist background interference and gesture occlusion.
2. **GRU-LSTM combined sequence learning network**: Draws on the design idea of lightweight models in robotic perception, combines the fast convergence of GRU and the high fitting accuracy of LSTM, and reduces the computational complexity by 28% while improving the recognition accuracy, which is

more suitable for real-time application scenarios.

3. **Adaptive gesture segmentation method**: Combines adaptive threshold segmentation and motion gesture segmentation, solves the problems of traditional segmentation methods being sensitive to light and missegmenting in skin-like backgrounds, and provides a high-quality gesture region for subsequent feature extraction.

In addition, the proposed method innovatively applies the **robust adaptive control theory** of autonomous systems to dynamic gesture recognition—by designing a multi-module adaptive adjustment mechanism (e.g., adaptive threshold segmentation for light changes, adaptive feature filtering for background interference), the model has strong adaptability to complex and uncertain real-world scenarios, which is different from the existing gesture recognition methods that only focus on model structure optimization.

4.2 Contribution analysis of core modules

The ablation experiment results show that the CBAM attention module and GRU-LSTM combined network are the two core modules affecting the model performance, and their respective contributions are as follows:

1. **CBAM attention module**: The module only improves the recognition accuracy by 0.30% on the basis of the Inception module, but it significantly improves the model's feature extraction efficiency—by filtering the key features of gestures and suppressing background noise, the number of effective features is reduced by 40%, and the inference speed is increased by 5 fps. This is consistent with the research results of adaptive feature filtering in robotic manipulation systems, which proves that the attention mechanism can effectively improve the efficiency of feature extraction in complex scenes.
2. **GRU-LSTM combined network**: The module is the biggest contributor to the accuracy improvement (5.10% higher than the single LSTM), which is because the GRU layer effectively solves the gradient vanishing problem of RNN in long sequence learning, and the two LSTM layers further enhance the model's ability to capture the temporal dependence of dynamic gesture data. The experimental results show that the combined network is more suitable for dynamic gesture sequence learning than a single LSTM/GRU, and its performance is better than the existing LSTM-GRU hybrid models in Ilham2024.

4.3 Robustness analysis under complex environmental changes

To verify the model's robustness, the proposed method is tested on the SWT dataset under different environmental changes (light intensity, shooting angle, skin tone), and the results show that:

1. **Light intensity change**: The recognition accuracy only decreases by 2.3% from normal light to strong light, and 2.8% to weak light, which is because the adaptive threshold segmentation method can adjust the segmentation threshold according to the light intensity, and the CBAM module can filter the light noise in the image features.
2. **Shooting angle change**: The recognition accuracy is 97.1% at a 30° side angle and 95.8% at a 60° side angle, which is because the MediaPipe model can accurately extract hand joint features at different angles, and the dual-mode feature fusion makes up for the deficiency of image feature distortion at large angles.
3. **Skin tone change**: The recognition accuracy of different skin tones (white, yellow, dark yellow) is above 97%, which proves that the proposed method has no skin tone bias and is suitable for different populations.

In addition, the sensitivity analysis of segmentation errors shows that when the segmentation IoU decreases from 95% to 85%, the recognition accuracy only decreases by 3.2%, which indicates that the model has low sensitivity to segmentation errors and strong robustness in practical applications.

4.4 Limitations of the proposed method

Although the proposed method achieves good performance on the SWT dataset, it still has the following limitations, which are also the key points of subsequent improvement:

1. **Limited dataset diversity**: The SWT dataset only contains the seven-step handwashing gesture, and the number of subjects and gesture types is relatively small, which leads to the model's poor generalization ability to other dynamic gestures (e.g., industrial control gestures, sign language gestures).
2. **Low CPU inference speed**: The model's inference speed on the CPU platform is only 15 fps, which is difficult to meet the real-time requirements of embedded devices (e.g., medical wearable devices) with limited computing resources.
3. **Poor handling of severe occlusion**: The model's recognition accuracy of severely occluded gestures (occlusion rate >50%) is only 82.5%, which is because the MediaPipe model cannot accurately extract joint features under severe occlusion, leading to the loss of key local

features.

4. **No consideration of gesture speed changes**: The model uses a fixed sparse downsampling strategy (s=30), which cannot adapt to the speed changes of dynamic gestures (e.g., fast handwashing gestures will cause frame loss, and slow gestures will cause redundant information).

5 Conclusion and future work

5.1 Conclusion

Aiming at the problems of poor gesture segmentation, incomplete feature extraction, low recognition accuracy and poor real-time performance in complex background dynamic gesture recognition, this paper proposes a dual-mode feature fusion dynamic gesture recognition method based on GRU-LSTM. The main research conclusions and contributions of this paper are as follows:

1. **A robust gesture segmentation method is proposed**: By combining adaptive threshold segmentation and motion gesture segmentation, the method solves the problems of traditional segmentation methods being sensitive to light intensity and missegmenting in skin-like backgrounds, and achieves an average IoU of 91.4% and an average Dice of 92.5% on the SWT dataset, with a segmentation speed of 33.5 fps meeting real-time requirements.
2. **A dual-mode feature extraction and fusion framework is constructed**: The MediaPipe model is used to extract hand joint local features, and an improved CNN with Inception and CBAM modules is used to extract gesture image global features. The two features are fused by concatenation and dimension reduction, which makes up for the deficiency of single feature expression and improves the model's anti-interference ability.
3. **A GRU-LSTM combined sequence learning network is designed**: The network combines the fast convergence of GRU and the high fitting accuracy of LSTM, which reduces the computational complexity by 28% compared with a single LSTM and improves the fitting accuracy by 2.7% compared with a single GRU, effectively capturing the temporal dependence of dynamic gesture sequence data.
4. **A dual-mode feature combined classification method is designed**: Based on MLP and Softmax-SVM combined algorithm, the method selects the optimal classification result for each gesture frame, and achieves an average recognition accuracy of 97.73% on the SWT dataset (Precision=97.21%, Recall=96.89%, F1-score=97.05%), with an inference speed of 28 fps on the GPU platform, which is better than the existing SOTA methods.

The proposed method is verified on the self-constructed SWT dataset, and the experimental results show that the method has high recognition accuracy, strong robustness and good real-time performance, and can accurately segment and recognize the seven-step handwashing dynamic gestures in complex backgrounds. The method provides a new solution for dynamic gesture recognition in professional fields such as medical and health care, and the design idea of dual-mode feature fusion and lightweight sequence learning can be extended to other human-computer interaction fields such as smart home and industrial control.

5.2 Future work

In view of the limitations of the proposed method, the following future research directions are proposed to further improve the model's performance and generalization ability:

1. **Expand the dataset diversity**: Collect more dynamic gesture datasets (e.g., medical control gestures, sign language gestures) with more subjects, more gesture types and more complex backgrounds, and use data augmentation techniques (e.g., random cropping, rotation, light adjustment) to expand the dataset, improving the model's generalization ability.
2. **Optimize the model for lightweight**: Use model compression techniques (e.g., quantization, pruning, knowledge distillation) to compress the proposed model, reduce the number of model parameters and computational complexity, and improve the inference speed on CPU and embedded devices, realizing the deployment of the model on medical wearable devices.
3. **Improve the handling of severe occlusion**: Combine the depth camera to collect 3D gesture data, and use the 3D convolution network (3D-CNN) to extract spatial-temporal features of 3D gestures, making up for the loss of 2D image features under severe occlusion, and improving the recognition accuracy of severely occluded gestures.
4. **Design an adaptive downsampling strategy**: According to the speed of dynamic gestures, design an adaptive sparse downsampling strategy that can adjust the sampling length s in real time, avoiding frame loss of fast gestures and redundant information of slow gestures, and improving the model's adaptability to gesture speed changes.
5. **Fuse multi-modal sensor data**: Combine visual sensors with inertial sensors (e.g., accelerometer, gyroscope) to collect gesture motion data, and fuse visual features and inertial features to further improve the model's robustness in extreme complex backgrounds (e.g., dark environment, heavy occlusion).
6. **Apply to practical medical systems**: Integrate the proposed method into the medical

handwashing supervision system of hospitals, realize the real-time recognition and standard evaluation of the seven-step handwashing technique, and provide technical support for hospital infection control.

References

[1] A. Barbhuiya, R. K. Karsh, and R. Jain. A convolutional neural network and classical moments-based feature fusion model for gesture recognition. Multimed. Syst., 28(5):1779–1792, 2022. https://doi.org/10.1007/s00530-022-00951-5

[2] A. Ilham, and I. Nurtanio. Applying LSTM and GRU methods to recognize and interpret hand gestures, poses, and face-based sign language in real time. J. Adv. Comput. Intell. Intell. Inform., 28(2):265–272, 2024. https://doi.org/10.20965/jaciii.2024.p0265

[3] Nimbekar, Y. S. Dinesh, A. Gautam, V. Hunsigida, A. R. Nali, and A. Acharyya. Reconfigurable VLSI design architecture for deep learning established forelimb and hindlimb gesture recognition for rehabilitation application. IEEE Access, 11:70061–70070, 2023. https://doi.org/10.1109/access.2023.3293422

[4] F. Zhang. Human–Computer Interactive Gesture Feature Capture and Recognition in Virtual Reality. Ergonomics Des., 29(2):19–25, 2021. https://doi.org/10.1177/1064804620924133

[5] H. Haroon, S. Altaf, Z. U. Rehman, M. W. Soomro, and S. Iqbal. Human hand gesture identification framework using SIFT and knowledge-level technique. ETRI J., 45(6):1022–1034, 2023. https://doi.org/10.4218/etrij.2022-0281

[6] H. Mahmud, M. M. Morshed, and M. K. Hasan. Quantized depth image and skeleton-based multimodal dynamic hand gesture recognition. Vis. Comput., 40(1):11–25, 2024. https://doi.org/10.1007/s00371-022-02762-1

[7] H. Zhang, H. Qu, L. Teng, and C. Y. Tang. LSTM-MSA: A novel deep learning model with dual-stage attention mechanisms forearm EMG-based hand gesture recognition. IEEE Trans. Neural Syst. Rehabil. Eng., 31:4749–4759, 2023. https://doi.org/10.1109/tnsre.2023.3336865

[8] J. Ni, Y. Wang, G. Tang, W. Cao, and S. X. Yang. A lightweight GRU-based gesture recognition model for skeleton dynamic graphs. Multimed. Tools Appl., 83(27):70545–70570, 2024. https://doi.org/10.1007/s11042-024-18313-w

[9] J. Zhang, and X. Zeng. Multi-touch gesture recognition of Braille input based on Petri Net and RBF Net. Multimed. Tools Appl., 81(14):19395–19413, 2022. https://doi.org/10.1007/s11042-021-11156-9

[10] M. U. Rehman, F. Ahmed, M. A. Khan, U. Tariq, F. A. Alfouzan, N. M. Alzahrani, and J. Ahmad. Dynamic hand gesture recognition using 3D-CNN and LSTM networks. Comput. Mater. Contin., 70(3):4675–4690, 2022. https://doi.org/10.32604/cmc.2022.019586

[11] S. M. S. Shah, J. I. Khan, S. H. Abbas, and A. Ghani. Symmetric mean binary pattern-based Pakistan sign language recognition using multiclass support vector machines. Neural Comput. Appl., 35(1):949–972, 2023. https://doi.org/10.1007/s00521-022-07804-2

[12] W. Qi, S. E. Ovur, Z. Li, A. Marzullo, and R. Song. Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network. IEEE Robot. Autom. Lett., 6(3):6039–6045, 2021. https://doi.org/10.1109/lra.2021.3089999

[13] Jin, X. Ma, Z. Zhang, Z. Lian, and B. Wang. Interference-robust millimeter-wave radar-based dynamic hand gesture recognition using 2-d cnn-transformer networks. IEEE Internet Things J., 11(2):2741–2752, 2023. https://doi.org/10.1109/jiot.2023.3293092

[14] Yuanyuan, S. H. I., Yunan, L. I., Xiaolong, F. U., Miao, K., & Miao, Q. (2021). Review of dynamic gesture recognition. *Virtual Reality & Intelligent Hardware*, *3*(3), 183-206. https://doi.org/10.1016/j.vrih.2021.05.001

[15] Sayed, U., Bakheet, S., Mofaddel, M. A., & El-Zohry, Z. (2024). Robust Hand Gesture Recognition Using HOG Features and machine learning. *Sohag Journal of Sciences*, *9*(3), 226-233. DOI:10.21608/sjsci.2024.248702.1151

[16] Boulkroune, A., Boubellouta, A., Bouzeriba, A., & Zouari, F. (2025). Practical finite-time fuzzy synchronization of chaotic systems with non-integer orders: Two chattering-free approaches. *Journal of Systems Science and Systems Engineering*, *34*(3), 334-359. https://doi.org/10.1007/s11518-024-5635-7

[17] Rigatos, G., Abbaszadeh, M., Busawon, K., Dala, L., Pomares, J., & Zouari, F. (2024). Flatness-based control in successive loops for autonomous quadrotors. *Journal of Dynamic Systems, Measurement, and Control*, *146*(2), 024501. https://doi.org/10.1115/1.4063907

[18] Rigatos, G., Siano, P., Zouari, F., & Ademi, S. (2020). Nonlinear optimal control of autonomous submarines' diving. *Marine Systems & Ocean Technology*, *15*(1), 57-69. https://doi.org/10.1007/s40868-019-00070-3

[19] Rigatos, G., Busawon, K., Abbaszadeh, M., Pomares, J., Gao, Z., & Zouari, F. (2024, August). Flatness-based control in successive loops for dual-arm robotic manipulators. In *2024 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 793-798). IEEE. doi: 10.1109/CCTA60707.2024.10666567.

[20] Rigatos, G., Siano, P., Zouari, F., & Ademi, S. (2020). Nonlinear optimal control of autonomous submarines' diving. *Marine Systems & Ocean Technology*, *15*(1), 57-69. https://doi.org/10.1007/s40868-019-00070-3

[21] Zouari, F., Mahmud, M. (2026). Neural

Network-Based Robust Adaptive Output Feedback Control for MIMO Time-Varying Delay Systems. In: Mahmud, M., Pillay, N., Kaiser, M.S. (eds) Applications of Artificial Intelligence and Data Science. AAIDS 2024. Communications in Computer and Information Science, vol 2601. Springer, Cham. https://doi.org/10.1007/978-3-031-98498-3_5