Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065895

ARTICLE

Prediction of Assembly Intent for Human-Robot Collaboration Based on Video Analytics and Hidden Markov Model

Jing Qu[1], Yanmei Li[1,2], Changrong Liu[1], Wen Wang[1] and Weiping Fu[1,3,*]

[1]School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, 710048, China
[2]Liupanshan Laboratory, Yinchuan, 750021, China
[3]School of Engineering, Xi'an International University, Xi'an, 710077, China

*Corresponding Author: Weiping Fu. Email: weipingf@xaut.edu.cn

Received: 24 March 2025; Accepted: 19 May 2025; Published: 03 July 2025

ABSTRACT: Despite the gradual transformation of traditional manufacturing by the Human-Robot Collaboration Assembly (HRCA), challenges remain in the robot's ability to understand and predict human assembly intentions. This study aims to enhance the robot's comprehension and prediction capabilities of operator assembly intentions by capturing and analyzing operator behavior and movements. We propose a video feature extraction method based on the Temporal Shift Module Network (TSM-ResNet50) to extract spatiotemporal features from assembly videos and differentiate various assembly actions using feature differences between video frames. Furthermore, we construct an action recognition and segmentation model based on the Refined-Multi-Scale Temporal Convolutional Network (Refined-MS-TCN) to identify assembly action intervals and accurately acquire action categories. Experiments on our self-built reducer assembly action dataset demonstrate that our network can classify assembly actions frame by frame, achieving an accuracy rate of 83%. Additionally, we develop a Hidden Markov Model (HMM) integrated with assembly task constraints to predict operator assembly intentions based on the probability transition matrix and assembly task constraints. The experimental results show that our method for predicting operator assembly intentions can achieve an accuracy of 90.6%, which is a 13.3% improvement over the HMM without task constraints.

KEYWORDS: Human-robot collaboration assembly; assembly intent prediction; video feature extraction; action recognition and segmentation; HMM

1 Introduction

With the continuous progress of robotics technology, the Human-Robot Collaboration Assembly (HRCA) production mode propels the traditional manufacturing production mode toward intelligence. Among them, the prediction of the collaborative robot's behavior intention based on the recognition of the operator's assembly movements is an important research direction of HRCA, which is of great significance for improving the flexibility and adaptability of the production system while ensuring the overall production efficiency of the assembly to achieve seamless collaboration.

Traditional action recognition methods mainly use manual feature-based methods [1–3]. Most of these methods need help with problems of complex preprocessing, slow speed, poor stability, and feature selection that significantly affect the accuracy and efficiency of action recognition. Deep learning methods effectively avoid these problems due to their superior learning capabilities, and researchers have focused on using deep learning for action recognition.

In the field of 3D Convolutional Neural Networks (3D CNNs), researchers have proposed a variety of network architectures to optimize video processing tasks, including spatiotemporal feature extraction and action classification. Chen et al. [4] utilized C3D network multi-frame inputs for end-to-end video data processing. The I3D model proposed by Carreira et al. [5] improves the ability to capture temporal information by extending 2D CNN weights to 3D. The SlowFast network of Feichtenhofer [6] balances capturing spatial and temporal information through a two-branch design. Zhu et al. [7] utilized an A3D network combined with adaptive mechanism optimization for spatiotemporal feature extraction. Methods such as X3D and TS-D3D [8,9] further explore the extension of the convolutional kernel and optimization of the network structure to improve action recognition accuracy. However, their computational complexity is high.

Some researchers [10–13] have investigated methods based on 2D convolutional neural networks to reduce computers' processing power while maintaining the accuracy of action video feature extraction. Strategies such as temporal interleaving [14], pyramidal structures [15], and differential networks [16] have also been used to improve the extraction of temporal features.

In summary, current deep learning methods have achieved good results in human action recognition. However, most of the current research focuses on recognizing daily behaviour actions. In the industrial environment, assembly actions are flexible and changeable and have temporal constraints between various actions during the assembly process, and the influence of the temporal factors of assembly actions needs to be considered in the research.

However, most of the current research focuses on the recognition of daily behaviors. In industrial environments, assembly actions are flexible and variable, and there are time constraints between various actions in the assembly process, which require considering the influence of the time factor of assembly actions. Assembly action recognition belongs to the application field of action recognition methods. Research on action recognition based on machine vision and deep learning has achieved a high accuracy in laboratory environments [17]. In recent years, scholars have begun to pay attention to the problem of action understanding in human-robot collaboration assembly scenarios, dividing continuous human activity streams into different parts through temporal action segmentation methods, each part corresponding to a semantically meaningful action, and realizing the segmentation of simple assembly actions [18,19]. Jones et al. [20] extended the fine-grained settings of action recognition for the assembly process, considering the relevance between actions and assembly objects, by unifying assembly actions and motion structures in a framework, realizing assembly action perception at the spatial detail level, and comprehensively achieving the task of assembly action recognition. Koch et al. [21] proposed an assembly progress estimation method based on Methods-Time-Measurement (MTM) and action recognition for a multivariate assembly process, combining the results of skeleton data action recognition to complete progress estimation. The existing methods still have significant gaps in the hierarchical action representation of complex assembly. Therefore, the research on assembly action recognition in industrial scenarios is still in the early exploration stage, and the key challenges such as robustness, fine-grained action segmentation, and real-time requirements in complex environments have not been fully resolved. Video streams are usually transmitted during the video data collection process, and the assembly process combines multiple assembly actions. The assembly process and actions are flexible and variable, and action recognition methods can only predict single-action videos. The manual editing and processing of video data undoubtedly causes a huge workload. Therefore, it is necessary to have methods that can directly process uncut, long-time sequence video action segmentation and recognition for the assembly process.

According to the different processing processes and methods of the action segmentation task, existing methods can be divided into direct segmentation methods (Direct Segmentation) and indirect segmentation

methods (Indirect Segmentation). Direct segmentation methods separate the segmentation process from the action recognition process, usually adopting a two-stage framework: first, generate coarse-grained action boundaries based on time series or sensor data, and then classify the segmented segments [22]; indirect segmentation methods achieve synchronous processing of action segmentation and recognition through end-to-end models, with typical representatives such as joint modeling methods based on temporal convolutional networks (TCN) or Transformers [23]. The performance differences between the two methods are mainly reflected in the long-term dependency modeling capability of complex action sequences. Shou et al. [24] achieve the segmentation of video actions by using three 3D convolutional networks based on segmentation to respectively realize the candidate segments of actions possibly contained in the nominated video, the initialization of the action classification model positioning network, and the fine-tuning network to locate each action, and finally recognize the nominated action segments. However, in implementing action nomination, it is necessary to separate non-action intervals between multiple actions in the video. Otherwise, it is difficult to accurately locate the start and end moments of the action intervals, which in turn affects the subsequent extraction and recognition of actions. Therefore, the action nomination method is unsuitable for continuous operations in the assembly process.

Current research has made progress in identifying individual assembly actions, but there is still little research on the recognition and segmentation of continuous assembly actions in assembly scenes [25]. Therefore, the focus of this study is to address the recognition and segmentation of continuous assembly actions, which is also a key and prerequisite issue for understanding and predicting the operator's assembly intention in the process of human-robot collaborative assembly.

Research on human intention prediction mainly focuses on sensor technology [26,27] and data-driven deep learning methods [28–32]. Behavioral intention recognition methods based on inertial measurement units (IMUs), pressure sensors, and other physiological signals can effectively solve the recognition blind spots of the visual modality in occlusion scenarios through multi-sensor data fusion. However, methods based on deep learning rely on large-scale offline data training to cover potential patterns in the high-dimensional state-action space (such as the sensitivity of the model to data distribution shifts) [33], and the prediction process is limited in interpretability due to the black-box characteristics [34]. Studies have shown that the performance of data-driven methods is strongly positively correlated with the amount of training data, but it is difficult to reach the causal reasoning ability of humans under a small number of samples [35]. The Hidden Markov Model (HMM) [36] requires a smaller sample size for effective learning and prediction. Therefore, this paper proposes an innovative HMM algorithm that incorporates assembly task constraints to improve the interpretability of prediction results by introducing task-specific constraints. At the same time, the algorithm also includes an incremental learning mechanism so that the model can flexibly adjust the parameters to better adapt to the needs of intention prediction under different assembly sequences. This research not only enriches the technical means of assembly intent prediction but also provides strong support for improving the intelligence of human-machine collaboration. The main contributions of this work can be summarized as follows:

(1) Feature extraction of assembly action video clips. To solve the problem of significant computation and poor real-time performance of 3D-CNN-based video feature extraction network, this paper adopts ResNet50 [37] as the backbone network, introduces Temporal Shift Module (TSM) [10] into it, and constructs a TSM-ResNet50 network structure for extracting the spatiotemporal features of video clips to reduce the amount of data and the subsequent model computation.

(2) Recognition and segmentation of continuous assembly actions. In this paper, MS-TCN++ [38] is adopted as the backbone network, and the Adaptive Temporal Fusion module (ATFM) and the Efficient Channel Attention module (ECA) [39] are embedded into the MS-TCN++ network to construct the

Refined-MS-TCN network. The model predicts the action categories and intervals existing in the unedited assembly video, verifies the method's feasibility through instance validation, and conducts model comparison experiments to verify the method's accuracy.

(3) Aiming at the problem that deep learning-based operator assembly intention prediction methods require a large amount of training data and the prediction process is not sufficiently interpretable, an operator assembly intention prediction model based on Hidden Markov Model (HMM) with assembly task constraints is proposed. Meanwhile, an incremental learning method is introduced to adjust the model parameters to adapt the intention prediction under different assembly sequences.

2 Materials and Methods

2.1 Overview of the Modeext Layout

In this paper, an operator assembly intent prediction method based on video analysis with Hidden Markov Models is proposed to enable the prediction of the operator's next assembly intent. An overview of the framework of the operator assembly intention prediction system for human-machine collaborative reducer assembly scenarios is given in Fig. 1. It mainly consists of two parts, assembly action recognition, segmentation, and operator assembly intention prediction, which are described as follows. It primarily consists of the following steps: (1) extraction of action features from the assembly action dataset; (2) recognition and segmentation of successive assembly actions based on the extracted action features; and (3) prediction of operator intention based on the obtained assembly action sequence. These three parts are described accordingly below.

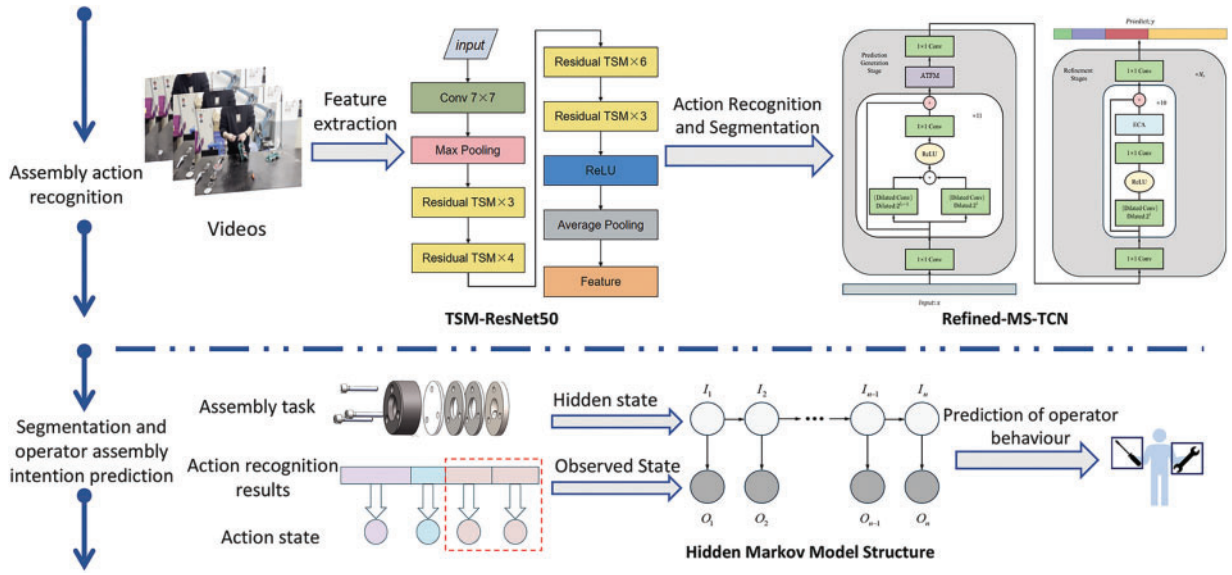

Figure 1: Technical roadmap of paper research

2.2 Based on TSM-ResNet50 Video Feature Extraction Network

Currently, video spatiotemporal feature extraction is commonly used in 3D Convolutional Neural Networks (3D CNN), which have high computational complexity and poor real-time performance. To achieve a good compromise between computational efficiency and performance, this paper constructs a video feature extraction network based on TSM-ResNet50 for extracting spatio-temporal features in

assembled videos, and the framework diagram of the feature extraction network is shown in Fig. 2. In TSM, T denotes the temporal dimension of the extracted feature maps, C denotes the channel dimension of the feature maps, and H and W represent the height and width of the feature map.

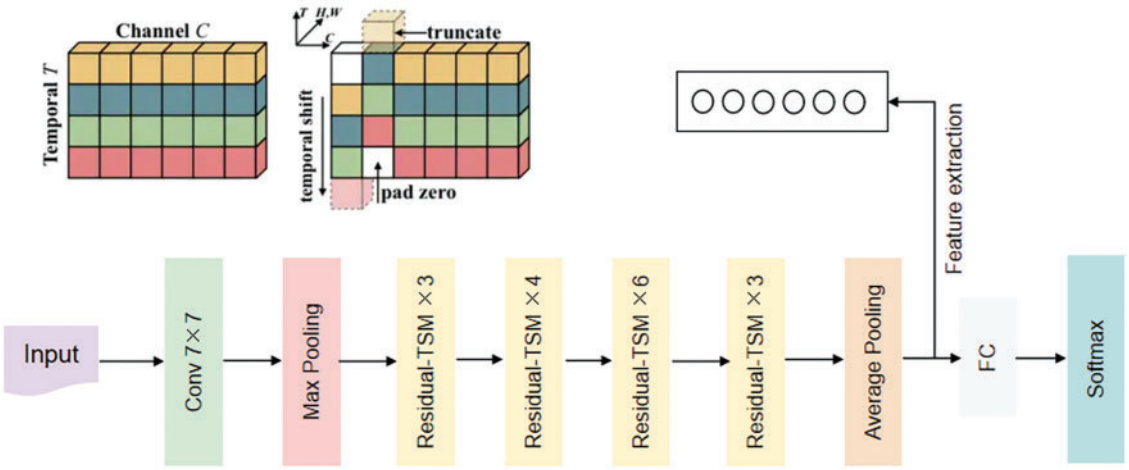

Figure 2: TSM-ResNet50 network structure

The core idea of TSM is to shift part of the passes of the feature map back and forth along the time dimension so that the feature map at a given moment contains the information of the neighboring frames, which can express the temporal characteristics of the video. In addition, since shifting the passes will damage the spatial information of the video, TSM is inserted into the main path of the ResNet50 network to maintain the integrity of the original spatial data through the residual connection of the ResNet50 network. This paper is based on the ResNet50 network as the backbone network of the feature extraction network. The TSM-ResNet50 video feature extraction network structure is constructed. Among them, the Residual Temporal Shift module (Residual-TSM) has two main paths, in which the time-shift residual mapping path is used for time-shift and convolution operations, and the other branching judgment path can retain the features before the time-shift operation so that the overall accuracy of the network can be ensured while modeling the temporal information.

The core idea of TSM is to shift part of the passes of the feature map back and forth along the time dimension so that the feature map at a given moment contains the information of the neighboring frames, which can express the temporal characteristics of the video. In addition, since shifting the passes will damage the spatial information of the video, TSM is inserted into the main path of the ResNet50 network to maintain the integrity of the original spatial data through the residual connection of the ResNet50 network. This paper is based on the ResNet50 network as the backbone network of the feature extraction network. The TSM-ResNet50 video feature extraction network structure is constructed. Among them, the Residual Temporal Shift module (Residual-TSM) has two main paths, in which the time-shift residual mapping path is used for time-shift and convolution operations, and the other branching judgment path can retain the features before the time-shift operation so that the overall accuracy of the network can be ensured while modeling the temporal information.

Since the feature maps extracted by ResNet50 can only represent spatial features, while TSM performs partial channel shift operations on the feature maps of video frames at time t, the feature map at time t contains information from the video frames at time $t-1$ and $t+1$. Although the shift operation affects the feature information of the 2D spatial dimension, the original 2D spatial features can be preserved by

the residual operation; since the shift operation does not increase the computation of the network, the performance of 3D CNN can be achieved with the complexity of 2D CNN.

To improve the network training performance, this paper adopts a pre-training + fine-tuning strategy to train the TSM-ResNet50 network. The ResNet50 network is first pre-trained using the ImageNet dataset [40], and the trained network parameters are used as the initial TSM-ResNet50 network model parameters. Then, the initial model parameters were fine-tuned using the self-constructed assembly action video data set to obtain the optimized TSM-ResNet50 network model. During the fine-tuning process, for each video clip, eight frames were randomly extracted from it, and then the frames were normalized to a size of 224 × 224 and fed into the network for training.

Since the temporal field of the network expands by 2 bits after each insertion of TSM, the final temporal field of the whole framework will be significant. To obtain sufficient temporal information, this paper extracts the fully connected (FC) layer in the TSM-ResNet50 network before the $M \times 2048$ dimensional output to characterize the temporal features of the video clip, in which M is the number of input video frames, and 2048 is the spatiotemporal feature dimension of the video frames.

2.3 Based on Refined-MS-TCN Action Recognition and Segmentation Network

2.3.1 Refined-MS-TCN Network Architecture

In this paper, MS-TCN++ is used as the backbone network, and the adaptive temporal fusion module (ATFM) and the efficient channel attention module (ECA) are embedded into the MS-TCN++ network to construct the refined MS-TCN network. These two modules complement the refined MS-TCN++ network, and the overall architecture of action recognition and segmentation is shown in Fig. 3.

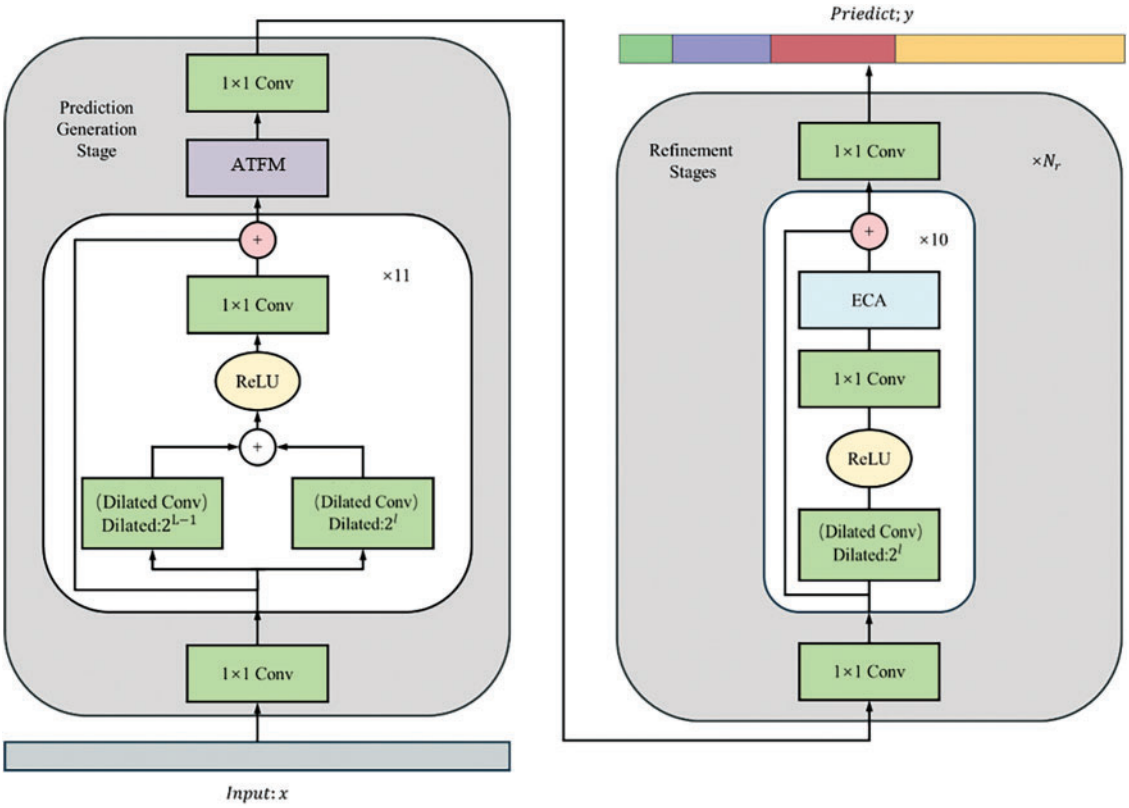

Figure 3: Refined-MS-TCN network schematic diagram

The ATFM module is a response to the AUGFPN [41]. The Adaptive Spatial Fusion Module (ASFM) in the target detection network is adapted to obtain the structures shown in Fig. 4. By introducing the attention mechanism, it can adaptively weight the layered temporal context features, thus effectively aggregating multi-scale temporal information and reducing the interference of irrelevant features.

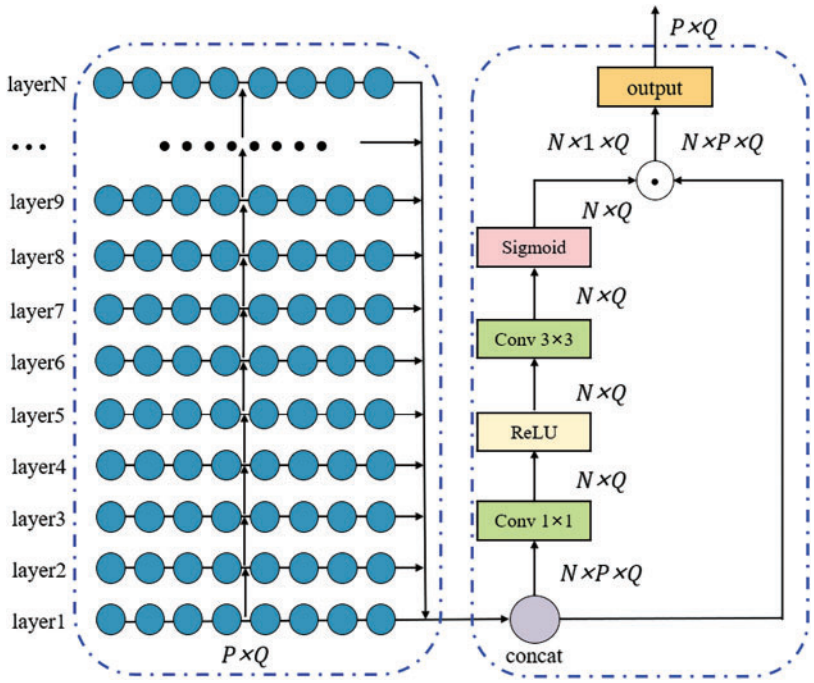

Figure 4: Adaptive temporal fusion module

ATFM extracts multi-scale temporal context features in the Prereduction Generation Stage of the Refined-MS-TCN network and splices these features as inputs; subsequently, ATFM assigns a temporal weight to each scale of temporal features. Finally, ATFM generates temporal context information with high-level semantics by aggregating these scales. This is done as shown below:

$$o_n = f_n\left(o_{n-1}\right), \tag{1}$$

$$H\left(x\right) = E\left(g\left(o_1, o_2, \ldots, o_n\right)\right), n \in N, \tag{2}$$

where $o_1 = x$ denotes the input features (video features) of the network; o_n denotes the output of the n dilated convolutional layer; f_n denotes the convolution operation of the n dilated convolutional layer; H denotes the output result of ATFM; E denotes the adaptive weighted fusion operator; g denotes the cascade operator; N denotes the number of inflated convolutional layers in each stage. The temporal feature size for all scales is $P \times Q$, where P is the number of feature channels and Q is the number of frames.

The ATFM module consists of a 1 × 1 convolution structure, a ReLU activation function, a 3 × 3 convolution structure, a Sigmoid activation function, and a matrix dot product operation. As a 'plug-and-play' module, ATFM has a small number of parameters, so its integration into the network does not increase the computational burden. In Refined-MS-TCN, only the inputs of the first stage of the network are the features extracted from the video, while the inputs of the subsequent stages are the outputs of the previous stage.

The ECA module finely tunes the feature channels through the weighting mechanism, which can adaptively enhance the features that have a more significant impact on the classification results and suppress the features that have little or no effect. The structure of this mechanism is shown in Fig. 5.

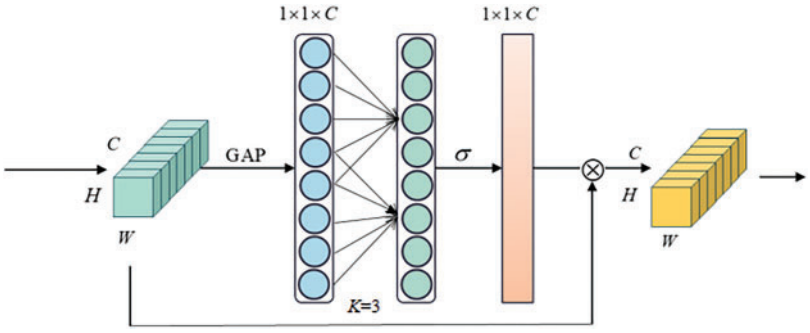

Figure 5: ECA channel attention mechanism

The ECA module compresses the feature X in the time dimension using Global Average Pooling (GAP), which converts the time dimension channel into a $1 \times C$ vector with a global perceptual field. It represents the international distribution of responses across the feature channels. To adjust the weight of each channel feature, a 1×1 convolution is used in the cross-channel feature interaction network. According to the size of the channel dimension, the convolution kernel size is set to 3 in this paper, where the size of the convolution kernel represents the range of cross-channel interaction. The convolution result is then activated using the Sigmoid activation function to obtain the weights ω of each channel, channel weights ω are multiplied by feature vectors to output weighted features η, as shown follows:

$$\omega = \sigma\left(C1D_3\left(GAP\left(X\right)\right)\right), \tag{3}$$

$$\eta = X \otimes \omega. \tag{4}$$

The ECA module has lower complexity and comparable performance to other attention mechanisms. This is because the ECA module does not require dimensionality reduction when performing inter-channel interactions, thus reducing the loss of feature information. At the same time, the network can learn each channel's weight coefficients and therefore distinguish each channel's features more effectively.

2.3.2 Loss Function Design

For the loss function, MS-TCN++ uses a combination of multi-categorical cross-entropy loss L_{cls} and smoothing loss L_{T-MSE}. In this paper, we use the Gaussian similarity weight loss function L_{GT-MSE} [42] to replace the loss function L_{T-MSE} of MS-TCN++ as part of the loss function. The GS-TMSE function penalizes neighboring frames with large differences with a smaller weight, ensuring a smooth action transition. The loss function for each stage of Refined-MS-TCN is:

$$L_s = L_{cls} + \lambda L_{GT-MSE}, \tag{5}$$

where L_{cls} is the Cross-Entropy Loss at each stage, L_{GT-MSE} is the Gaussian weighted smoothing loss at each stage, and the network hyperparameter λ is 0.15.

The classification loss L_{cls} and the Gaussian similarity weighted loss L_{GT-MSE} are defined as follows:

$$L_{cls} = \frac{1}{T}\sum_{t} -log(y_t, c) \tag{6}$$

$$L_{GS-TMSE} = \frac{1}{TN}\sum_{t,c} \tilde{\Delta}_{t,c}^2 exp\left(-\frac{\|x_t - x_{t-1}\|^2}{2}\right)\tilde{\Delta}_{t,c}^2$$

$$\tilde{\Delta}_{t,c} = \begin{cases} \Delta_{t,c}^2: & \Delta_{t,c} \le \tau \\ \tau: & otherwise \end{cases}$$

$$\Delta_{t,c} = |logy_{t,c} - logy_{t-1,c}|$$

where T is the length of the video, N is the number of categories, and $y_{t,c}$ is the probability that the category is c at time t. The x_t is the input feature of frame t, and the parameter τ is set to 4.

2.4 Operator Assembly Intent Predictive Model Incorporating HMM and Assembly Task Constraints

2.4.1 Modelling Stage

In the Human-Robot Collaboration Assembly process, the operator's next assembly task can be taken as the operator's assembly intention. In this paper, the assembly tasks in the assembly process are taken as the hidden states of the HMM, and the probabilistic transfer relationship between assembly tasks is obtained through the state transfer probability matrix A of the HMM. The assembly tasks are not observable to the robot, and what the robot can observe is the assembly action state sequence obtained by the action recognition and segmentation model. The assembly tasks are treated as the hidden states of the HMM, while the observed assembly action states are treated as the observed states of the HMM. Observed states are the states that can be directly observed externally, and they are linked to the internal hidden states through the HMM. The assembly tasks and observed assembly action states during assembly are deg-7ned as shown in Table 1.

Table 1: Assembly task and observed assembly action state of Hidden Markov model

Assembly task	Assembly task meaning	Observed assembly action state	Observed assembly action state meaning
q_1	Assembly of oil level indicator	v_1	q_1 action observed
q_2	Assembly of the oil retainer ring at the end of the input shaft cover	v_2	q_2 action observed
q_3	Assembly of input shaft through-cap end bearing	v_3	q_3 action observed
q_4	Assembly of the input shaft cover	v_4	q_4 action observed
q_5	Assembly of the oil retainer ring at the end of the input shaft cover	v_5	q_5 action observed
q_6	Assembly of input shaft mullion end bearing	v_6	q_6 action observed
q_7	Assembly of input shaft adjusting ring and cover	v_7	q_7 action observed
q_8	Assembly of the input shaft	v_8	q_8 action observed

(Continued)

Table 1 (continued)

Assembly task	Assembly task meaning	Observed assembly action state	Observed assembly action state meaning
q_9	Assembly of output shaft large gear	v_9	q_9 action observed
q_{10}	Assembly of output shaft sleeve	v_{10}	q_{10} action observed
q_{11}	Assembly of output shaft through-cap end bearing	v_{11}	q_{11} action observed
q_{12}	Assembly of output shaft cover	v_{12}	q_{12} action observed
q_{13}	Assembly of output shaft mullion end bearing	v_{13}	q_{13} action observed
q_{14}	Assembly of output shaft adjusting ring and cover	v_{14}	q_{14} action observed
q_{15}	Assembly of output shafts	v_{15}	q_{15} action observed
q_{16}	Assembly of the upper case cover of the gearbox	v_{16}	q_{16} action observed

The relationship between the observed assembly action states and the assembly task can be obtained through the firing matrix B of the HMM. Since the results obtained by the action recognition model are not necessarily accurate, the observed assembly action states are not necessarily correct. In this paper, we mitigate the impact of action recognition errors on subsequent intent prediction by corresponding a hidden state to multiple observed states. Using one hidden state to correspond to various observed states, even if there is an error in the observed assembly action state, the HMM can use the Viterbi algorithm to obtain the correct hidden state and then correctly predict the operator's assembly intention.

2.4.2 Incremental Learning Stage

The accuracy of the operator's prediction of assembly intent is closely related to the parameters of the HMM. For an untrained sequence of observed assembly action states, a state transfer probability matrix in the HMM corresponding to a hidden state transfer probability of 0 will not correctly predict the intention of the operator.

In this paper uses incremental learning approach to dynamically adapt the parameters of the HMM. The data for incremental learning is obtained from the action recognition and segmentation model. After each assembly, the assembly action sequences obtained from the action recognition and segmentation model are recorded, and a set of observed assembly action state sequences are obtained using an action sequence processing algorithm and assembly action state extraction. Assume that after the t-th assembly, the parameters of the Hidden Markov Model (HMM) are $\lambda_{t-1} = (A_{t-1}, B_{t-1}, \pi_{t-1})$, which represents the parameters of the HMM after the $(t-1)$-th assembly is completed. After the t-th assembly, the parameters of the HMM model trained by the BW algorithm using the recorded observed assembly action state sequence O_t are denoted as $\lambda_t^{'} = (A_t^{'}, B_t^{'}, \pi_t^{'})$. The calculation method for generating new parameters $\lambda_t = (A_t, B_t, \pi_t)$ using incremental learning in the HMM is as follows:

$$\lambda_t = \begin{cases} \lambda_{t-1} + \eta \times \lambda_t^{'}, & \log P(O_t|\lambda') > \varepsilon \\ \lambda_{t-1}, & \text{otherwise} \end{cases}, \tag{7}$$

where η denotes the learning rate, which represents the trade-off between learning new knowledge and forgetting old knowledge; larger η indicates that the HMM is more inclined to forget old knowledge and is better able to adapt to new environments, while smaller η indicates that the HMM is more inclined to retain historical experience. Suppose the likelihood probability $\log P(O|\lambda)$ is smaller than the set threshold ε. In that case, it can be considered that this test sample has high confidence and can be used for the updating operation of the HMM parameters (in this paper, the parameters of the firing matrix B are not updated, and B is only related to the performance of the assembly action recognition model). The learning rate η is set to 0.2. The threshold ε is obtained by weighting the $\log P(O|\lambda)$ obtained by using the BW algorithm on a single training sample, and its computational equation is as follows:

$$\varepsilon = K \times \frac{1}{N} \sum_{i=1}^{N} \log(O_i|\lambda_i), \tag{8}$$

where N denotes the number of training samples, since $\log(O_i|\lambda_i)$ is the likelihood probability derived on the HMM generated from the training samples, the probability value will achieve a larger value, which is adjusted using the scaling factor K to set the threshold ε more objectively. In subsequent incremental learning experiments, K was set to 0.8.

2.4.3 Prediction Stage

The inputs to the HMM prediction stage are the observed assembly action state sequence O and the parameter λ of the HMM, the corresponding hidden state sequence is obtained by using the Viterbi algorithm, and the hidden state q_i with the largest transfer probability of the current hidden state q_j is found by the probability transfer matrix A. The assembly task represented by q_j is the prediction result of the operator's assembly intention.

During the assembly process of the reducer, assembly task constraints refer to the fact that certain assembly tasks must be completed before others can be carried out, and all assembly tasks must be performed in a specific order. Some assembly tasks must also be carried out in a specific sequence. For example, the assembly of the output shaft, the assembly of the input shaft, and other assembly tasks must be completed before the assembly of the top cover of the gear reducer. In this paper, the assembly task to be completed before the assembly task is called the preliminary assembly task, the gearbox assembly process of the assembly task of the initial assembly task as shown in Table 2. In addition, each assembly task in the transmission assembly process must be performed only once in the whole assembly process. Once an assembly task is completed, it should not be used as the result of predicting the operator's assembly intention.

Table 2: The precedent task of the assembly task

Assembly task	Pre-assembly tasks
q_1	Null
q_2	Null
q_3	Null
q_4	q_2q_3
q_5	Null
q_6	q_5
q_7	$q_2q_3q_4q_5q_6q_8$
q_8	$q_2q_3q_4q_5q_6$

(Continued)

Table 2 (continued)

Assembly task	Pre-assembly tasks
q_9	Null
q_{10}	q_9
q_{11}	Null
q_{12}	q_{11}
q_{13}	$q_9 q_{10}$
q_{14}	$q_9 q_{10} q_{11} q_{12} q_{13} q_{15}$
q_{15}	$q_9 q_{10} q_{11} q_{12} q_{13}$
q_{16}	$q_2 q_3 q_4 q_5 q_6 q_7 q_8 q_9 q_{10} q_{11} q_{12} q_{13} q_{14} q_{15}$

In predicting the operator's assembly intention, the sequence of observed assembly action states can be obtained by the action recognition model and assembly action sequence processing method. Then the corresponding sequence of hidden states and the current hidden state (assembly task) can be obtained by the Viterbi algorithm through the state transfer probability matrix A; we can get the transfer probability of each hidden state corresponding to the current hidden state, and then predict the operator's next assembly intention. The operator's assembly intention is then predicted. In the intention prediction process, assembly task constraints are added to determine whether the predicted assembly task (operator intention) has been completed and its predecessor assembly task has been completed. Suppose the previous assembly task still needs to be completed or the predicted assembly task has been completed. In that case, the intention prediction result is changed to the assembly task with the next highest transfer probability as the operator assembly intention prediction result, and the cycle is repeated until the final intention prediction result is obtained intention prediction result. The specific steps of the operator assembly intention prediction method based on HMM and assembly task constraints are shown in Fig. 6.

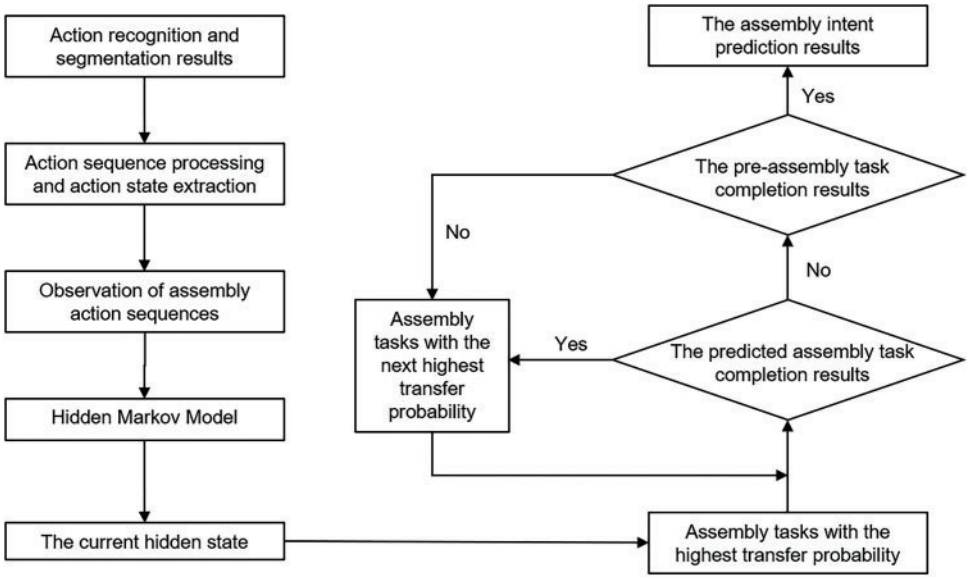

Figure 6: Operator assembly intent prediction model based on HMM and assembly task constraints

3 Experiment and Discussion

Experimental analyses are carried out using the captured reducer assembly video dataset, and a comparative model is built to validate and evaluate the approach adopted in the model based on the effect of action segmentation and recognition in real assembly videos.

3.1 Dataset Introduction

3.1.1 Assemble Data Collection

In this study, a reducer assembly process is used as the source of the assembly action dataset. The assembly process of the reducer is collected using data acquisition equipment (Xiaomi 11, with a resolution of 108 MP for the primary camera), and a reducer assembly action dataset is constructed.

The dataset contains 79 videos, each with 16 assembly actions, with an average duration of about 3 s for each action. The video frame rate is 30 fps. The videos are recorded from the perspective of the main view. As indicated by the name of the dataset, these videos describe the assembly process of the reducer. To ensure the diversity of experimental data, 20 experimenters were invited to perform reducer assembly tasks, and the proficiency of each participant in the assembly process was different. During the recording process, participants repeated multiple times to complete the assembly process of the assembly body. A video data contains a complete and continuous assembly process. Due to the different proficiency of participants in the assembly process, the duration of completing the assembly action will vary, so the overall data can meet the diversity requirements.

3.1.2 Assembly Action Division

Based on the analysis of the reducer's assembly process, this paper associates the assembly actions with the various components of the reducer. Different assembly actions are divided through the assembly of other components, each assembly action including the entire process of assembling the corresponding components, such as the assembly action of the upper cover of the reducer, which includes the gripping, placement, and fastening of the upper cover of the reducer. Due to the absence of reducer components, this paper does not involve the assembly actions of components such as the reducer sight cover and reducer oil plug. Since there are repeated components in the reducer, such as oil baffles, deep groove ball bearings 6204, etc., this paper distinguishes the components with the same name by differentiating the input shaft and output shaft and the through cover installation end of the shaft and the blind cover installation end.

This article divides explicitly the assembly actions in the reducer assembly process into 16 categories, as shown in Fig. 7. Among these 16 assembly actions, some are relatively special. In the reducer assembly process, the installation of small covers, reflectors, and oil hole gaskets is continuous, so the assembly action of the oil level indicator includes the installation of small covers, oil hole indicators, reflectors, and oil hole gaskets and other components. The assembly of the input shaft and the assembly action of the input shaft are to install all the components on the shaft into the housing.

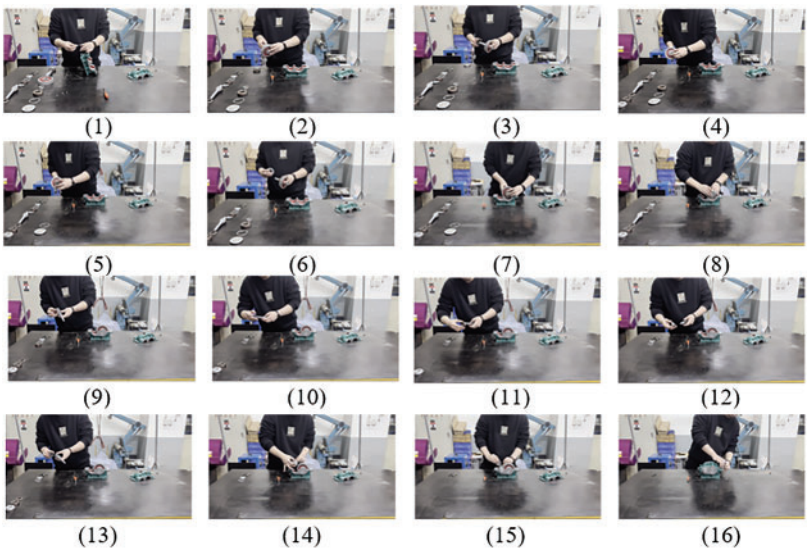

Figure 7: Reducer assembly action categories. **(1)** Assembling the oil level indicator; **(2)** assembling the output shaft gear; **(3)** assembling the output shaft end cover bearing; **(4)** assembling the output shaft sleeve; **(5)** assembling the output shaft end cover bearing; **(6)** assembling the output shaft end cover; **(7)** assembling the output shaft; **(8)** assembling the output shaft adjustment ring and end cover; **(9)** assembling the input shaft end cover oil damper; **(10)** assembling the input shaft end cover oil damper; **(11)** assembling the input shaft end cover bearing; **(12)** assembling the input shaft end cover; **(13)** assembling the input shaft end cover bearing; **(14)** assembling the input shaft; **(15)** assembling the input shaft adjustment ring and end cover; **(16)** assembling the reducer top cover

3.1.3 Preprocessing and Clipping of Assembly Action Dataset

In this paper, high-definition video data was obtained from the assembly process of the subjects' reducers, with a resolution of 1920 × 1080 per frame and a frame rate of 30 fps. However, high-resolution images will increase the burden of network computation. In order to optimize computational efficiency, this section adjusts the resolution of each frame image to 224 × 224. Considering that the action between adjacent two frames of the video often changes little, one image frame is selected every two image frames, thereby reducing the frame rate of the video to 10 fps. This data processing method retains the critical information of the video data while reducing the computational burden, making the motion features more prominent and conducive to subsequent video feature extraction tasks.

This section edits the complete assembly video, dividing it into multiple single assembly action videos according to each assembly action's start and end times, to train the video feature extraction network.

3.2 Video Feature Extraction Experiment

3.2.1 Evaluation Indicators

Euclidean distance measures the absolute distance between two points in multidimensional space. The formula for Euclidean distance in n-dimensional space is:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)}. \tag{9}$$

Cosine similarity is a measure of the similarity between two vectors by measuring the cosine of the angle in the direction of the two vectors. The closer the cosine similarity of two vectors is to 1, the more similar

they are, and the closer the similarity is to 0, the more different they are. For vector $X = (x_1, x_2, \ldots, x_n)^T$ and vector $Y = (y_1, y_2, \ldots, y_n)^T$, the cosine similarity formula is:

$$cos(\theta) = \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}}. \tag{10}$$

3.2.2 Experiment Dataset

The self-constructed complete reducer assembly action video dataset is randomly divided into two subsets, with 80% used for training and 20% for testing, both containing assembly action records of different personnel.

3.2.3 Experimental Results and Analyses

In this paper, the TSM-ResNet50 video-based feature extraction network and the I3D feature extraction network are used to extract features from the complete video of transmission assembly, respectively. The average similarity between the TSM-ResNet50 network and the I3D network for extracting frame features of the same assembly action category and frame features of different assembly action categories is also compared. The results of the comparison of the network feature extraction results are shown in Table 3. By calculating the similarity between the feature vectors extracted from each frame, the distribution of the feature data in the time dimension still retains temporal characteristics, indicating that the TSM-ResNet50 feature extraction network has effectively preserved the temporal information in the video, and the feature extraction model is adequate.

Table 3: Network feature extraction effectiveness

Feature extraction method	Mean Euclidean distance for the same action frames	Mean Euclidean distance for the different action frames	Mean cosine similarity for the same Action Frames	Mean cosine similarity for the different action frames
I3D	3.42	8.84	0.992	0.762
TSM-ResNet50	2.86	8.96	0.993	0.772

3.3 Experiments on Continuous Assembly Action Recognition and Segmentation

3.3.1 Evaluation Indicators

In this paper, to verify the effect of Refined-MS-TCN action recognition and segmentation network on the recognition and segmentation of assembly action, we choose to adopt *Acc* (frame-wise accuracy), and *F1-score* as the main evaluation criteria of this assembly action recognition and segmentation experiment.

3.3.2 Experimental Setup

The action recognition and segmentation network based on Refined-MS-TCN has 4 phases, the first phase is the prediction generation phase, which includes 11 DDL modules as well as the ATFM module, and the last 3 phases are the prediction tuning phases, each including 10 inflated convolutional layers based on residual connectivity and the ECA module. Dropout is used after each layer and the probability is set to 0.5, the number of channels in each convolutional layer is 64, the size of the convolutional kernel is 3, and λ is set

to 0.15. In all experiments, the Adam optimizer is used with a learning rate of 0.0005, and no weight decay is set.

3.3.3 Experiment Dataset

To demonstrate the effectiveness of the ATFM module and the ECA module introduced in this paper, network structure splitting experiments were performed on the self-constructed assembly dataset, using MS-TCN++ as a reference. The input features of the self-constructed assembly action dataset use the N × M × 2048 dimensional features of the full assembly video extracted with the TSM-ResNet50 video feature extraction network, where N is the number of videos, and M is the number of video frames. Each assembly action has three basic attributes: the category of the action, the starting moment, and the ending moment. The action recognition network needs to input action labels for supervised learning during action learning, so it is necessary to label the actions contained in each video in the dataset. When performing data annotation, it is also required to base it on the frame number of the image to complete the annotation of the various attributes of the actions in the video data and generate the annotation file of the dataset. Taking the collected reducer assembly video dataset as an example, the annotation method is shown in Fig. 8.

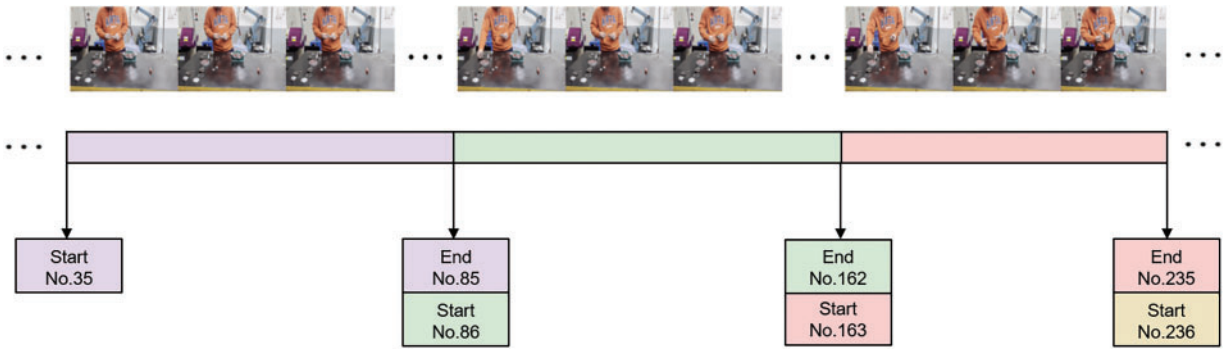

Figure 8: Dataset annotation method

3.3.4 Experimental Results and Analyses

As shown in Table 4, for the task of recognition and segmentation of assembly operations, the refined MS-TCN network performs better than MS-TCN++ in all evaluation metrics, demonstrating the effectiveness of the improvements in this paper. Integrating the ATFM module into the MS-TCN++ network can improve its performance. This is because ATFM weights the multi-scale features in the first stage of the network, which can better capture the dynamic changes in different time scales, produce more accurate initial results, and provide a good initial value for the subsequent correction stage of the network.

Table 4: Experimental results of self-built assembly action dataset

Self-built assembly action dataset	F0.1	F0.25	F0.5	Acc
MS-TCN++ & ATFM	90.5	88.4	76.2	81.8
MS-TCN++ & ECA	88.6	87.2	74.8	79.2
MSTCN++ & ATFM & GS-TMSE	90.7	88.6	76.2	82.0
MS-TCN++ & ECA & GS-TMSE	88.8	85.7	76.0	80.1
Ours	92.9	92.0	82.9	83.0

We compare the proposed model with the state-of-the-art method on the self-built assembly dataset. As shown in Table 5, our model's *F*1-*score* and frame-wise accuracy (*Acc*) are both better than the state-of-the-art methods.

Table 5: Comparison with the state-of-the-art on self-built assembly action dataset

Self-built assembly action dataset	F0.1	F0.25	F0.5	Acc
MS-TCN++ [38]	88.2	86.2	75.9	80.7
AsFormer [43]	77.2	72.5	77.9	51.8
ICC [44]	43.9	47.2	46.8	25.2
Ours	92.9	92.0	82.9	83.0

The advantage of combining the ATFM and ECA modules in the MS-TCN++ network is that they complement each other to improve the overall network performance. Although the ECA module alone may not improve the network's performance, the two can complement each other when combined with the ATFM module. The ATFM module provides more accurate initial features to the ECA module, allowing the ECA to weight the features better, further improving the accuracy of the final detection. The use of the GS-TMSE loss function has little impact on network performance. The primary purpose of introducing the GS-TMSE loss function is to ensure that the network can partition different assembly action intervals and mitigate the effect of similar assembly actions on the results.

To more intuitively display the segmentation results, this section visualizes the action recognition and segmentation results of the self-built assembly action dataset videos, as shown in Fig. 9. From the figure, it can be seen that the Refined-MS-TCN used in this chapter can segment assembly actions well and correctly classify assembly actions. However, there are still some things that need to be corrected, including missed recognition of assembly actions and assembly actions with durations significantly different from the actual situation. The missed recognition is because some assembly actions have short durations and are very similar to adjacent assembly actions, so multiple consecutive assembly actions are recognized as a single assembly action.

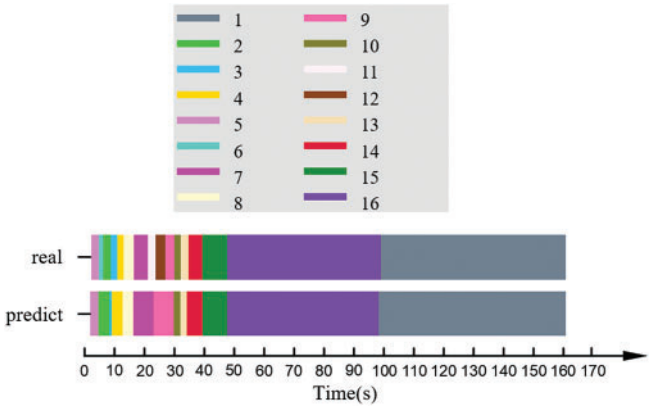

Figure 9: Visualization of assembly action recognition and segmentation results

3.4 *Experiment to Predict Operator Assembly Intent*

3.4.1 HMM Model

In this paper, 40 reducer assembly videos are processed using the action recognition and segmentation network, and the assembly task sequences and the observed assembly action state sequences are extracted from them, the number of which is 16, and they are matched as the training data of the HMM, and the match is shown in Fig. 10. Different numbers in the figure represent different assembly tasks and observed assembly action states, respectively, and the part marked by the red box indicates that the assembly tasks do not match with the observed assembly action states, and this relationship is reflected in the firing matrix B of the HMM.

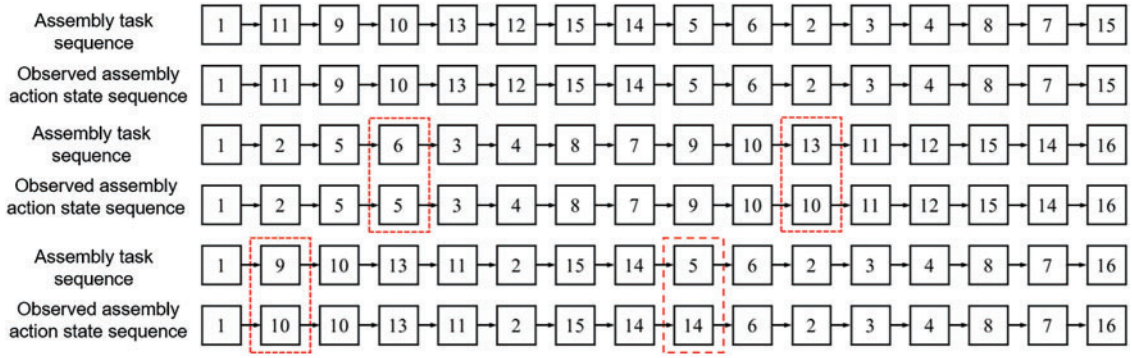

Figure 10: Model training data

Using these 40 sequences, the state transfer probability matrix A, the firing matrix B and the initial state probability distribution matrix π of the Hidden Markov Model (HMM) are approximated by simple statistical frequency calculations. The results are shown in Tables A1–A3 in Appendix A.

3.4.2 Experiment Dataset

To verify the effectiveness of this paper's operator assembly intent prediction method based on HMM with assembly task constraints, an actual Human-Robot Collaboration Assembly scenario is built in this section, as shown in Fig. 11.

Figure 11: Human-robot collaborative assembly scenario

During the experiment, 5 subjects were set up, each completing one complete reducer assembly. The subjects were required to freely choose the assembly sequence during the assembly process without any assembly errors. The complete assembly action sequence was extracted using an action recognition and segmentation model and processed using an action sequence processing method. The processed assembly sequence was segmented, as shown in Fig. 12, ensuring that the assembly action at the segmentation point had been completed for some time and the following assembly action had yet to start. This paper defines 16 assembly actions, which can be segmented into 15 assembly action sequences, resulting in 75 test data. After segmentation, assembly action state extraction was performed on them. The test data were input into HMM and HMM with assembly task constraints to test the effect of assembly intention prediction. HMM uses the initial probability matrix to predict the operator's first assembly task, fixed at q_1 ($\pi = 0.4$ without incremental training, with the maximum probability value).

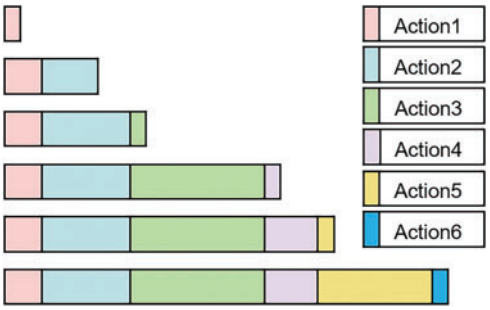

Figure 12: The test data segmentation method

3.4.3 Experimental Results and Analyses

The test data were fed into the HMM with the addition of assembly task constraints to examine the effectiveness of the assembly intention prediction. The HMM predicts the operator's first assembly task using an initial probability matrix π fixed at q_1 without incremental training. The experimental results are shown in Table 6, where the prediction accuracy is the ratio of the number of experimental samples correctly predicted to the number of samples in all experiments. The HMM with the addition of the assembly task constraints can achieve a prediction accuracy of 90.6%, which is an improvement of 13.3% compared to the method using the HMM.

Table 6: Operator assembly intent prediction accuracy

Method	Predictive accuracy
HMM	77.3%
The HMM with the addition of the assembly task constraints	90.6%

4 Conclusions

In this paper, we use an innovative approach to assembly intent prediction. Based on the HMM and assembly task constraints, an efficient operator assembly intent prediction model is developed. First, the interpretability of the HMM prediction results is improved by incorporating assembly task constraints; in particular, the impact of action recognition errors on intent prediction is mitigated by introducing task-specific constraints. Then, an incremental learning mechanism is incorporated to allow the model to flexibly

adjust the parameters to better meet the needs of intention prediction under different assembly sequences. In terms of feature extraction from assembly action video clips, the Temporal Shift Module (TSM) is introduced into the ResNet50 network to extract the spatio-temporal features of the video clips to alleviate the subsequent computational burden. An enhanced MS-TCN network is used for assembly action detection and segmentation, and the accuracy of action detection is improved by introducing the Adaptive Temporal Fusion Module (ATFM) and the Efficient Channel Attention (ECA) module. Experiments on our self-built reducer assembly action dataset demonstrate that our network can classify assembly actions frame by frame, achieving an accuracy rate of 83%. Finally, an operator assembly intention prediction system based on video analysis and HMM is constructed. The experimental results show that our method for predicting operator assembly intentions can achieve an accuracy of 91.7%, which is a 13.3% improvement over the HMM without task constraints. This paper's method improves the intelligence level of human-robot collaboration assembly and provides a new approach to assembly intention prediction. Although the model shows high prediction accuracy in experiments, further optimization is needed to improve the generalization ability and real-time performance of the model in practical applications. Future work will focus on lightweight and optimizing the model to include more real-world assembly scenarios and further improve the accuracy and efficiency of the prediction.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Jing Qu and Changrong Liu; methodology, Jing Qu, Yanmei Li and Changrong Liu; investigation, Changrong Liu; software, Jing Qu and Yanmei Li; formal analysis, Jing Qu, Yanmei Li and Changrong Liu; data curation, Jing Qu, Yanmei Li and Changrong Liu; writing—original draft preparation, Jing Qu and Changrong Liu; writing—review and editing, Jing Qu and Weiping Fu; supervision, Wen Wang; project administration, Weiping Fu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Weiping Fu, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A

Table A1: Initial state probability distribution matrix

State	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}
0.4	0.15	0	0	0.2	0	0	0	0.25	0	0	0	0	0	0	0	0

Table A2: State transition probability matrix

State	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}
q_1	0	0.375	0	0	0.125	0	0	0	0.375	0	0.125	0	0	0	0	0

(Continued)

Table A2 (continued)

State	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}
q_2	0	0	0.95	0	0.05	0	0	0	0	0	0	0	0	0	0	0
q_3	0	0	0	0.9	0.1	0	0	0	0	0	0	0	0	0	0	0
q_4	0	0	0	0	0.35	0	0	0.65	0	0	0	0	0	0	0	0
q_5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
q_6	0	0.5	0.05	0.1	0	0	0	0.35	0	0	0	0	0	0	0	0
q_7	0	0	0	0	0	0	0	0	0.35	0	0.2	0	0	0	0	0.45
q_8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
q_9	0	0	0	0	0	0	0	0	0	0.95	0.05	0	0	0	0	0
q_{10}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
q_{11}	0	0	0	0	0	0	0	0	0.05	0	0	0.95	0	0	0	0
q_{12}	0	0	0	0	0	0	0	0	0.2	0.05	0	0	0	0	0.75	0
q_{13}	0	0	0	0	0	0	0	0	0	0	0.7	0.05	0	0	0.25	0
q_{14}	0	0.2	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0.55
q_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
q_{16}	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A3: Emission matrix

State	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	q_{16}
q_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_2	0	0.925	0.075	0	0	0	0	0	0	0	0	0	0	0	0	0
q_3	0	0	0.8	0.2	0	0	0	0	0	0	0	0	0	0	0	0
q_4	0	0	0	0.975	0	0	0	0	0	0	0	0	0	0	0	0
q_5	0	0.025	0	0	0.85	0.1	0	0	0	0	0	0	0	0.25	0	0
q_6	0	0	0	0	0.1	0.9	0	0	0	0	0	0	0	0	0	0
q_7	0	0	0	0	0	0	0.8	0.2	0	0	0	0	0	0	0	0
q_8	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0.1	0
q_9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
q_{10}	0	0	0	0	0	0	0	0	0.1	0.9	0	0	0	0	0	0
q_{11}	0	0	0	0	0	0	0	0	0	0	0.85	0.15	0	0	0	0
q_{12}	0	0	0	0	0	0	0	0	0	0	0.025	0.975	0	0	0	0
q_{13}	0	0	0	0	0	0	0	0	0	0	0.15	0	0.85	0	0	0
q_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
q_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0.85	0
q_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

References

1. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); 2005 Jun 20–25; San Diego, CA, USA. p. 886–93. doi:10.1109/CVPR.2005.177.

2. Chaudhry R, Ravichandran A, Hager G, Vidal R. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. p. 1932–9. doi:10.1109/CVPR.2009.5206821.
3. Wang H, Schmid C. Action recognition with improved trajectories. In: 2013 IEEE International Conference on Computer Vision; 2013 Dec 1–8; Sydney, NSW, Australia. p. 3551–8. doi:10.1109/ICCV.2013.441.
4. Chen J, Wang M, Jiang S, Huang B, Sun H. Learning spatiotemporal features for video semantic segmentation using 3D convolutional neural networks. In: 2022 6th International Symposium on Computer Science and Intelligent Control (ISCSIC); 2022 Nov 11–13; Beijing, China. p. 55–62. doi:10.1109/ISCSIC57216.2022.00023.
5. Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 4724–33. doi:10.1109/CVPR.2017.502.
6. Feichtenhofer C, Fan H, Malik J, He K. SlowFast networks for video recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 6201–10. doi:10.1109/iccv.2019.00630.
7. Zhu S, Yang T, Mendieta M, Chen C. A3D: adaptive 3D networks for video action recognition. arXiv:2011.12384. 2020.
8. Feichtenhofer C. X3D: expanding architectures for efficient video recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 200–10. doi:10.1109/cvpr42600.2020.00028.
9. Yang M, Guo Y, Zhou F, Yang Z. TS-D3D: a novel two-stream model for action recognition. In: 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML); 2022 Oct 28–30; Xi'an, China. p. 179–82. doi:10.1109/ICICML57342.2022.10009839.
10. Lin J, Gan C, Han S. TSM: temporal shift module for efficient video understanding. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 7082–92. doi:10.1109/iccv.2019.00718.
11. Shao H, Qian S, Liu Y. Temporal interlacing network. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York, NY, USA. p. 11966–73.
12. Jiang B, Wang M, Gan W, Wu W, Yan J. STM: spatiotemporal and motion encoding for action recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 2000–9. doi:10.1109/iccv.2019.00209.
13. Yang C, Xu Y, Shi J, Dai B, Zhou B. Temporal pyramid network for action recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 588–97. doi:10.1109/cvpr42600.2020.00067.
14. Wang L, Tong Z, Ji B, Wu G. TDN: temporal difference networks for efficient action recognition. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 1895–904. doi:10.1109/CVPR46437.2021.00193.
15. Chen Y, Ge H, Liu Y, Cai X, Sun L. AGPN: action granularity pyramid network for video action recognition. IEEE Trans Circuits Syst Video Technol. 2023;33(8):3912–23. doi:10.1109/TCSVT.2023.3235522.
16. Ryu S, Hong S, Lee S. Making TSM better: preserving foundational philosophy for efficient action recognition. ICT Express. 2024;10(3):570–5. doi:10.1016/j.icte.2023.12.004.
17. Chen C, Zhang C, Wang T, Li D, Guo Y, Zhao Z, et al. Monitoring of assembly process using deep learning technology. Sensors. 2020;20(15):4208. doi:10.3390/s20154208.
18. Caccavale R, Saveriano M, Finzi A, Lee D. Kinesthetic teaching and attentional supervision of structured tasks in human-robot interaction. Auton Rob. 2019;43(6):1291–307. doi:10.1007/s10514-018-9706-9.
19. Mukherjee D, Gupta K, Chang LH, Najjaran H. A survey of robot learning strategies for human-robot collaboration in industrial settings. Robot Comput Integr Manuf. 2022;73(2):102231. doi:10.1016/j.rcim.2021.102231.
20. Jones JD, Cortesa C, Shelton A, Landau B, Khudanpur S, Hager GD. Fine-grained activity recognition for assembly videos. IEEE Robot Autom Lett. 2021;6(2):3728–35. doi:10.1109/LRA.2021.3064149.

21. Koch J, Büsch L, Gomse M, Schüppstuhl T. A methods-time-measurement based approach to enable action recognition for multi-variant assembly in human-robot collaboration. Procedia CIRP. 2022;106:233–8. doi:10.1016/j.procir.2022.02.184.
22. Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal convolutional networks for action segmentation and detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 1003–12. doi:10.1109/CVPR.2017.113.
23. Kuehne H, Arslan A, Serre T. The language of actions: recovering the syntax and semantics of goal-directed human activities. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA. p. 780–7. doi:10.1109/CVPR.2014.105.
24. Shou Z, Wang D, Chang SF. Temporal action localization in untrimmed videos via multi-stage CNNs. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 1049–58. doi:10.1109/CVPR.2016.119.
25. Wang P, Liu H, Wang L, Gao RX. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann. 2018;67(1):17–20. doi:10.1016/j.cirp.2018.04.066.
26. Liu Y, Liu X, Wang Z, Yang X, Wang X. Improving performance of human action intent recognition: analysis of gait recognition machine learning algorithms and optimal combination with inertial measurement units. Comput Biol Med. 2023;163(9):107192. doi:10.1016/j.compbiomed.2023.107192.
27. Liu Y, Liu X, Zhu Q, Chen Y, Yang Y, Xie H, et al. Adaptive detection in real-time gait analysis through the dynamic gait event identifier. Bioengineering. 2024;11(8):806. doi:10.3390/bioengineering11080806.
28. Zhang J, Wang P, Gao RX. Hybrid machine learning for human action recognition and prediction in assembly. Robot Comput Integr Manuf. 2021;72:102184. doi:10.1016/j.rcim.2021.102184.
29. Zhang Z, Peng G, Wang W, Chen Y, Jia Y, Liu S. Prediction-based human-robot collaboration in assembly tasks using a learning from demonstration model. Sensors. 2022;22(11):4279. doi:10.3390/s22114279.
30. Zhang Y, Ding K, Hui J, Lv J, Zhou X, Zheng P. Human-object integrated assembly intention recognition for context-aware human-robot collaborative assembly. Adv Eng Inform. 2022;54(11):101792. doi:10.1016/j.aei.2022.101792.
31. Liu Z, Liu Q, Xu W, Liu Z, Zhou Z, Chen J. Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing. Procedia CIRP. 2019;83(9):272–8. doi:10.1016/j.procir.2019.04.080.
32. Li S, Zheng P, Fan J, Wang L. Toward proactive human-robot collaborative assembly: a multimodal transfer-learning-enabled action prediction approach. IEEE Trans Ind Electron. 2022;69(8):8579–88. doi:10.1109/TIE.2021.3105977.
33. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44. doi:10.1038/nature14539.
34. Marcus G. Deep learning: a critical appraisal. arXiv:1801.00631. 2018.
35. Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. Building machines that learn and think like people. Behav Brain Sci. 2017;40:e253. doi:10.1017/S0140525X16001837.
36. Zhang X, Tian S, Liang X, Zheng M, Behdad S. Early prediction of human intention for human-robot collaboration using transformer network. J Comput Inf Sci Eng. 2024;24(5):051003. doi:10.1115/1.4064258.
37. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8. doi:10.1109/CVPR.2016.90.
38. Li S, Farha YA, Liu Y, Cheng MM, Gall J. MS-TCN++: multi-stage temporal convolutional network for action segmentation. IEEE Trans Pattern Anal Mach Intell. 2023;45(6):6647–58. doi:10.1109/TPAMI.2020.3021756.
39. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q. ECA-net: efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 11531–9. doi:10.1109/cvpr42600.2020.01155.
40. Deng J, Dong W, Socher R, Li LJ, Kai L, Li FF. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. p. 248–55. doi:10.1109/CVPR.2009.5206848.

41. Guo C, Fan B, Zhang Q, Xiang S, Pan C. AugFPN: improving multi-scale feature learning for object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 12592–601. doi:10.1109/cvpr42600.2020.01261.
42. Ishikawa Y, Kasai S, Aoki Y, Kataoka H. Alleviating over-segmentation errors by detecting action boundaries. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV); 2021 Jan 3–8; Waikoloa, HI, USA. p. 2321–30. doi:10.1109/wacv48630.2021.00237.
43. Yi F, Wen H, Jiang T. ASFormer: transformer for action segmentation. arXiv:2110.08568. 2021.
44. Singhania D, Rahaman R, Yao A. Iterative contrast-classify for semi-supervised temporal action segmentation. Proc AAAI Conf Artif Intell. 2022;36(2):2262–70. doi:10.1609/aaai.v36i2.20124.