Egyptian Informatics Journal 32 (2025) 100801

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Lightweight sign language intelligent recognition model based on improved R-C3D

Haofei Chen *, Chang'an Di

School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210000, China

ARTICLE INFO

Keywords:
Sign language
Intelligent recognition
Regional three-dimensional convolutional network
Depth separable convolution
Residual structure

ABSTRACT

The study proposes a continuous dynamic sign language recognition model based on an improved regional 3D convolutional network. A 3D convolutional network is taken as a special extraction sub-network, and the depth separable convolution is introduced into the 3D convolutional network to reduce computational costs. The inverted residual results are taken to avoid information loss issues. In addition, the pre-selection box size of the optimized region 3D convolutional network is shortened, and the action judgment threshold is increased to improve the action accuracy. The average accuracy of the improved 3D convolutional network was 44.2 %, which was higher than that of other types of feature extraction sub-networks. After reducing the pre-selection box, the average accuracy of the time suggestion sub-network increased from 41.6 % to 44.5 %. The loss value also decreased from 0.5 to 0.46. After increasing the action judgment threshold from 0.5 to 0.7, the loss value decreased from 0.58 to 0.17. The loss value of the 3D convolutional network in the entire improved area was only 0.15, the sign language recognition speed was 183 ms, and the average accuracy was 44.6 %, which was better than those of other sign language recognition schemes. The above results indicate that the improved regional 3D convolutional network can accurately and quickly recognize continuous sign language actions.

1. Introduction

According to the World Health Organization, there are approximately 65 to 70 million deaf and mute people worldwide, and China alone has 20.75 million people with hearing impairments. Sign language, as the main approach of communication for people with hearing impairments, mainly expresses meaning through gestures and arm movements. However, due to the low popularity of sign language in society, there are few people who can communicate with deaf mute people, which greatly hinders their work, education, and life. However, with the rapid advancement of Internet technology, sign language can be recognized through intelligent recognition systems, which greatly facilitates the life and communication of deaf mutes. Sign language recognition has two categories: static and dynamic sign language recognition. Among them, the former can only recognize gesture images at a certain point in time, which breaks the continuity of communication. Dynamic sign language recognition can greatly preserve the overall meaning of sign language and greatly weaken the isolation of static sign language vocabulary [1–3]. The current methods for dynamic sign language recognition include Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Long Short-term Memory networks (LSTM). RNN may encounter gradient vanishing or exploding problems when dealing with long sequence sign language data. LSTM requires a large amount of computing resources and has too many parameters. Although ordinary CNN can automatically extract information, it is hard to extract both spatial and temporal information simultaneously. Convolutional three-dimensional (C3D) can simultaneously extract these information, making it widely used in the action recognition. However, C3D has a complex structure, long training time, and high computational cost [4,5]. Therefore, to reduce computational complexity and ensure the accuracy of sign language recognition, a lightweight continuous dynamic sign language recognition model based on an optimized Region Convolutional 3D (R-C3D) network is proposed. The innovation lies in introducing depth separable convolution and inverted residual structures into the C3D network as feature extraction sub-networks. Secondly, the prediction box size is shortened and the action decision threshold is increased to reduce the prediction probability.

The research has four chapters. The first chapter is a related works, which briefly describes the relevant research on gesture recognition and

* Corresponding author.
E-mail address: chenhf19932024@163.com (H. Chen).

https://doi.org/10.1016/j.eij.2025.100801
Received 28 May 2025; Received in revised form 26 August 2025; Accepted 19 September 2025
Available online 16 October 2025

3D convolution. Chapter 2 is the continuous dynamic sign language recognition model based on improved R-C3D, which designs the lightweight continuous sign language recognition model based on improved R-C3D. Chapter 3 is the dynamic sign language detection results and analysis, which analyzes the various sub-networks and overall performance of the lightweight continuous sign language recognition model based on the optimized R-C3D. The final chapter is the conclusion, which summarizes the entire research.

2. Related works

Sign language is a communication method that expresses certain meanings through gestures. There is currently considerable research on gesture recognition in various fields. Saboo, Singha, and Laskar [6] built a gesture recognition method based on a hybrid classifier to remove irrelevant information in gesture recognition. This method detected and removed self-interconnect strokes through a two-step tilt angle system, and used the removed strokes as added features and suggested features. Compared with a single classifier, the accuracy of the hybrid classifier was improved by 2.66 %. Hu et al. [7] proposed an intelligent glove based on printed carbon nanotube graphene/PDMS strain sensors to achieve gesture recognition in intelligent communication. Customized artificial neural networks were used to achieve gesture classification. In addition, this recognition method also added an additional robotic hand, which could achieve simple communication with humans. It achieved an average precision of 97 % and an accuracy of 99.4 % for multiple gesture groups. Khattak et al. [8] developed a sign language recognition method based on surface electromyography signals and deep residual networks for gesture recognition in complex backgrounds. This method first processed surface electromyographic signals through Gaussian filtering and extracts their features, and then implemented sign language recognition through deep residual networks. The experimental results showed that the maximum accuracy of this method was 0.943, with a true positive rate of 0.929, which was superior to existing algorithms. Chen et al. [9] built a gesture recognition method based on surface electromyography decomposition technology and convolutional kernel compensation algorithm for real-time gesture recognition in multi-motion tasks. This method decoded the discharge of motion units across multiple motion tasks in a motion manner, and processed the decomposed motion segments using convolutional kernel compensation algorithm. The average classification accuracy reached 94.6 %. It had superiority in gesture recognition for multiple motion tasks. Lu, Le, and Kim [10] built a gesture recognition method based on a hybrid classifier and edge CNN for intelligent edge devices. This method utilized a hybrid classifier and majority voting scheme to balance accuracy and power consumption, and achieved interactive processing through edge CNN. It recognized 6 static gestures and 24 dynamic gestures, with an average accuracy of 87.25 %-95 % and 85.4 %-94.9 %.

3D convolutional networks, as a network that can capture both temporal and spatial information, have the advantages of processing volume data and knowledge transfer. Therefore, they are extensively applied in medical images, video classification, and behavior recognition. Z. Tao et al. built a lightweight feature extraction method based on 3D convolutional networks for restoring depth maps in RGB images. This method used 3D convolution to complete depth completion tasks. The designed lightweight 3D convolution had similar accuracy to standard 3D convolution, but the parameter size was the same as 2D convolution [11]. Wu et al. [12] designed a localization method based on 3DCNN for user localization in large-scale multi-input multi-output orthogonal frequency division multiplexing systems. This method replaced traditional fingerprints with an angle delay channel power matrix and extracted features from the angle delay channel power matrix through a 3D CNN to improve localization accuracy. This positioning method had higher positioning accuracy compared with traditional methods, which was robust to noise pollution. Kozlovskii and Popov [13] proposed a recognition model based on 3D CNN for protein peptide binding site recognition. This model used tensor-based spatial protein structure representation and obtained the probability score and coordinates of binding “hotspots” in the input structure through a 3D CNN. The model took no more than 1 s to analyze each protein. Hassan et al. [14] proposed a method based on deep learning and custom combustion layers to classify poultry audio signals. This method integrated digital audio signal processing, convolutional neural network (CNN), and innovative Burn Layer, enhancing the elasticity of the model to input signal changes by injecting controlled random noise during the training process. The results showed that the model exhibited excellent performance on six standard classification indicators, with a sensitivity of 96.77 %, specificity of 100.00 %, accuracy of 100.00 %, and Matthew correlation coefficient of 95.88 %, providing valuable insights for fields such as audio signal processing and animal health monitoring. Hassan et al. [15] proposed a method based on WaveNet and CNN-BiLSTM for detecting articulation disorders in patients with cerebral palsy and ALS. This method automatically classified speech disorders using the TORGO database and the Russian speech dataset, introduced a real-time adaptive framework to adapt to resource limited devices, and combined SHAP values and Grad CAM technology to enhance interpretability. The results showed that the accuracy, recall, and F-score of the WaveNet-based model were 0.92040, 0.91979, and 0.91965, respectively. The Matthews Correlation Coefficient (MCC) was 0.89320, which was superior to that of other models and demonstrated the potential for rapid and reliable detection of articulation disorders in multilingual and resource limited environments.

In summary, there are many achievements in research on gesture recognition, but most of them are focused on static gestures or individual dynamic gestures. As a means of communication, single dynamic sign language recognition cannot accurately reflect the expressed meaning. Therefore, it is necessary to recognize continuous dynamic sign language to alleviate communication barriers between deaf mute individuals and the normal population. At present, the development trend of sign language recognition is shifting from wearing complex data gloves to the mainstream direction based on computer vision. Compared with early sign language recognition methods based on wearable devices, sign language recognition methods based on computer vision do not rely on various complex sensor systems during data collection or method use. They only require ordinary RGB cameras or Kinect cameras, which reduces costs and optimizes user experience. However, there are still some shortcomings in current computer vision based sign language recognition methods, such as the susceptibility of images to lighting and complex backgrounds, and the difficulty in accurately segmenting continuous sign language. Therefore, the recognition accuracy still needs to be improved. Sign language recognition methods based on computer vision are shifting from the earliest machine learning methods to deep learning methods, and from theoretical research on the host side to convenient and mobile embedded deployment and implementation. As a widely used method for video classification and action recognition, 3D convolution can extract both temporal and spatial information simultaneously, but its computational cost is high. Therefore, to achieve accurate sign language recognition while reducing computational costs, a lightweight continuous dynamic sign language recognition method based on optimized R-C3D is proposed.

3. Continuous dynamic sign language recognition model based on improved R-C3D

This chapter conducts research on dynamic sign language recognition methods. A continuous dynamic sign language recognition model based on improved R-C3D is proposed. To improve the dynamic sign language recognition performance of the R-C3D network, the research will improve the feature extraction network and temporal prediction network of the R-C3D network.

3.1. Lightweight sign language dynamic recognition network based on improved C3D

Compared with 2DCNN, C3D can extract both temporal and spatial information simultaneously, making it more advantageous in dynamic sign language recognition. However, the computational cost and resource consumption of C3D exceed the2D convolution. Therefore, to reduce the computational cost of C3D, a lightweight C3D sign language recognition model is designed. The C3D structure is shown in Fig. 1.

From Fig. 1 C3D is composed of convolution layer, pool layer and full connection layer. Unlike 2D convolution, the convolution kernel used in C3D is 3D. The 3D convolution kernel can slide in the three dimensions of height, width and depth. Each slide can get a corresponding output. Therefore, C3D can extract temporal and spatial information. Different from the pooling operation of 2D convolution, the pooling operation of C3D also increases the depth on the basis of 2D pooling. When the pooling window slides every time, the pooling filter operator and characteristic data will perform an operation [16–18]. However, with the increase of convolution depth, C3D has some degradation problems. Therefore, to solve the above problems, 3D residual structure is introduced into C3D. The so-called residual structure refers to adding jump connections in the convolution layer, and performing a batch normalization and activation function operation after each 3D convolution layer. The characteristic layer obtained by residual structure is shown in Eq. (1).

$$P_i = \sigma(F(P_{i-1}) + P_{i-1}) \tag{1}$$

in Eq. (1), P_i represents the characteristic layer after the residual of layer i. $\sigma(.)$ indicates the ReLU activation function. $F(.)$ refers to the calculation of direct connection in the residual block. When the dimension of the current feature layer and the previous feature layer are different, the previous feature layer needs to be down-sampled. The ReLU activation function and direct connection are displayed in Eq. (2).

$$\begin{cases} \mathrm{Re}LU(x) = \max(0, x) \\ F(P) = BN(W_2(\sigma(BN(W_1(P))))) \end{cases} \tag{2}$$

in Eq. (2), $BN(.)$ represents the batch normalization. W_1 and W_2 represent the convolution calculation process of layer 1 and layer 2, respectively. Eq. (3) shows the batch normalization.

$$\begin{cases} BN(x_i) = \gamma \widehat{x}_i + \beta \\ \widehat{x}_i = \dfrac{x_i - \mu}{\delta} \\ \mu = \dfrac{1}{m}\sum_{i=1}^{m} x_i \\ \delta = \sqrt{\dfrac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2} \end{cases} \tag{3}$$

in Eq. (3), x_i represents the input value. γ represents the weight. β represents the deviation. δ represents the standard deviation. μ represents the mean. m represents the quantity of input data. Batch normalization can effectively reduce the training difficulty and improve its generalization ability. Although the above network can construct a dynamic sign language recognition model, its computational cost is high. Therefore, to reduce computational costs, the study introduces depth separable convolution in C3D. The depth-separable convolution is displayed in Fig. 2.

In Fig. 2, depth-separable convolution consists of two parts: depth convolution and point-wise convolution, which is essentially a decomposed convolution. The input data will first use a single convolution kernel for deep convolution, and then perform mixed convolution on the features obtained from deep convolution. This method can effectively reduce unnecessary calculations and parameters [19–21]. The computation required for this standard convolution is displayed in Eq. (4).

$$C_{std} = k \times k \times c_{in} \times c_{out} \times h \times w \tag{4}$$

in Eq. (4), C_{std} represents the amount of computation required for standard convolution. k signifies the convolution kernel size. c_{in} signifies the input channel number. c_{out} signifies the output channel number. h represents the input height. w represents the input width. When performing deep convolution, each input feature channel has a separate convolution layer, and the deep convolution is displayed in Eq. (5).

$$G(y, x, j) = \sum_{u=1}^{k}\sum_{u=1}^{k} K(u, u, j) \times I(y + u - 1, x + u - 1, j) \tag{5}$$

in Eq. (5), $G(y, x, j)$ signifies the result of deep convolution. (y, x, j) represents the input for deep convolution. $G(y, x, j)$ represents deep convolution kernel. I represents the input feature channel. The point-wise convolution is shown in Eq. (6).

$$O(y, x, l) = \sum_{j=1}^{c_{in}} M(y, x, j) \times P(j, l) \tag{6}$$

in Eq. (6), $O(y, x, l)$ represents the point-wise convolution result. (y, x, l) represents the input for point-wise convolution. M represents the input feature channel for point-wise convolution. P represents a point-wise convolution kernel. The computational cost of depth-separable convolution is displayed in Eq. (7).

$$C_{ds} = k \times k \times h \times w + c_{in} \times c_{out} \times h \times w \tag{7}$$

in Eq. (7), C_{ds} represents the computational cost of depth-separable convolution. Compared with standard convolution, the computational cost of depth-separable convolution is significantly reduced. To accurately measure the degree of reduction in computational complexity, a reduction factor is defined in Eq. (8).

$$\eta = \frac{C_{ds}}{C_{std}} = \frac{k^2 c_{in} hw + c_{in} c_{out} hw}{k^2 c_{in} c_{out} hw} = \frac{1}{c_{in}} + \frac{1}{k^2} \tag{8}$$

in Eq. (8), η represents the reduction factor. Although depth-separable convolution significantly reduces computational complexity, the model suffers from information loss due to the fact that depth-separable convolution kernels correspond to the number of feature channels. Therefore, to avoid information loss, the study borrows the idea of inverse residual structure. The inverted residual structure is similar to the residual structure, but the difference is that the inverted residual first performs point-convolution to enhance data dimensionality, and then

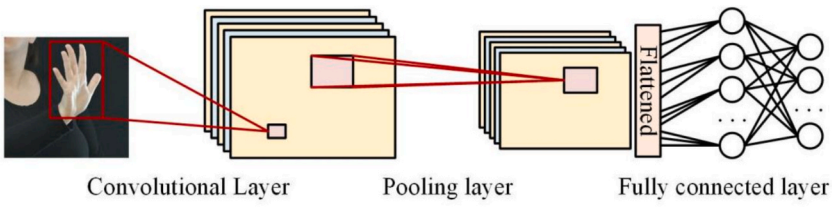


Fig. 1. Structure of C3D.

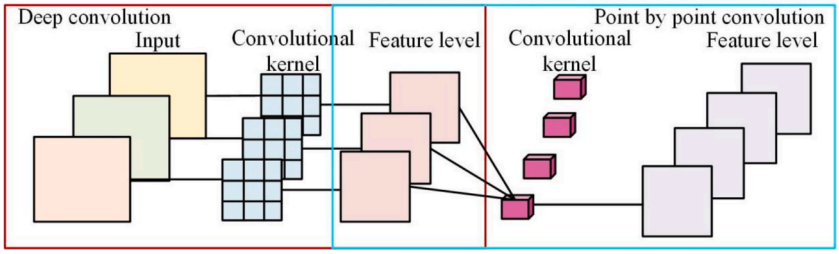


Fig. 2. Depth-separable convolution.

performs convolution and activation [22,23]. This approach can ensure that all operations are carried out in high dimensions, effectively reducing information loss. The characteristic of having fewer parameters for depth-separable convolution ensures that the complexity of the model remains low even in high-dimensional spaces. At this point, the increase in dimensionality for point-wise convolution is shown in Eq. (9).

$$P_i = \sigma\left(BN\left(W_p\left(P_{in}\right)\right)\right) \tag{9}$$

in Eq. (9), P_1 represents the data after dimensionality enhancement. W_p represents the convolution kernel parameter for point-wise convolution. P_{in} represents the input feature. To adapt to the changes in the model, the activation function is changed from ReLU to ReLU6, as displayed in Eq. (10).

$$\mathrm{Re}LU6(x) = \min(6, \max(0, x)) \tag{10}$$

After point-wise convolution, the dimension of channels is increased by 6 times. Therefore, it is necessary to perform point-wise convolution again after deep convolution to restore the channel dimension to its original state. At this point, the output feature is shown in Eq. (11).

$$P_{out} = P_{in} + BN\left(W_p\left(\sigma\left(BN\left(W_d\left(P_1\right)\right)\right)\right)\right) \tag{11}$$

in Eq. (11), P_{out} represents the output feature. W_d represents the convolution kernel parameter for deep convolution. From Eq. (11), after integrating deep convolution and inverse residual structure, the core is to reduce information loss while reducing computational complexity through a concatenation operation of "dimensionality enhancement – deep convolution – dimensionality reduction – skip connection". When the dimension of the output features and input features are different, down-sampling is required. At this point, lightweight dynamic sign language has been constructed by the network, and its structural parameters are displayed in Table 1.

According to Table 1, the size of the 3D convolution kernel is $3 \times 3 \times 3$. The pooling window step size is $2 \times 2 \times 2$. The first residual block has a step size of $1 \times 1 \times 1$. The remaining residual blocks have a step size of $2 \times 2 \times 2$.

Table 1
C3D lightweight dynamic sign language recognition network parameters.

Layer	Step length	Output
Input	/	$3 \times L \times 112 \times 112$
Conv $3 \times 3 \times 3$	$1 \times 1 \times 1$	$32 \times L \times 112 \times 112$
MaxPool	$2 \times 2 \times 2$	$32 \times \frac{L}{2} \times 56 \times 56$
Light-ResBlock	$1 \times 1 \times 1$	$32 \times \frac{L}{2} \times 28 \times 28$
Light-ResBlock	$2 \times 2 \times 2$	$64 \times \frac{L}{2} \times 14 \times 14$
Light-ResBlock	$2 \times 2 \times 2$	$128 \times \frac{L}{4} \times 7 \times 7$
Light-ResBlock	$2 \times 2 \times 2$	$256 \times \frac{L}{4} \times 4 \times 4$
AvgPool	/	$256 \times 1 \times 1 \times 1$
Linear	/	2

Note: L represents the number of dynamic gesture data frames for the input model.

3.2. *The lightweight sign language intelligent recognition model based on improved R-C3D*

Although the dynamic sign language recognition network mentioned above can recognize dynamic isolated sign language, it is hard to achieve continuous dynamic sign language recognition for unsegmented videos. Therefore, based on the C3D mentioned above, a lightweight continuous dynamic sign language recognition model based on optimized R-C3D is proposed. The R-C3D is displayed in Fig. 3.

In Fig. 3, R-C3D consists of three sub-networks: feature extraction, timing suggestion, and action classification. After using the C3D network to extract features from sign language videos, R-C3D generates a possible time range for actions and adjusts this time range. Finally, the model will select feature information within a time range to classify sign language. The feature extraction sub-network is mainly responsible for extracting and processing the temporal and spatial features of videos. To obtain more representative feature data, the improved C3D network used in the research is the feature extraction sub-network mentioned above. The improved C3D network introduces inverted residual structure and deep separation convolution, which can effectively ensure feature information and reduce computational complexity. However, due to the short duration and short intervals between sign language actions, continuous dynamic sign language recognition requires high accuracy in generating standard boxes. The traditional R-C3D time suggestion for each anchor point in the sub-network corresponds to a 10 scale, which is not sufficient to meet the requirements of continuous sign language dynamic recognition [24,25]. Meanwhile, when recognizing sign language, the pre-selection box is used to select the recognition object. When the generated pre-selection box is too large, the selected range will contain two or more actions, which will greatly interfere with accurate recognition. Therefore, larger pre-selection boxes are not suitable for recognizing sign language actions with shorter duration. Therefore, the study chooses to shrink the pre-selection box to meet the requirements of continuous dynamic sign language recognition. When the sampling camera samples 30 frames of images per second, the frame rate corresponding to each sign language action is approximately between 30 and 60 frames. When mapped to the input data for each unit length on the Base feature, it is 8 frames of images. The scale value after conversion should not exceed 8. Therefore, the scale used to generate standard candidate boxes is set to min scale {3, 4, 4.5, 5.5, 6, 6.5, 7, 7.5, and 8}. When the time suggestion sub-network obtains the basic features, it can then be convolved to obtain the offset information and foreground/background classification score of the prediction box. The time suggestion sub-network structure is shown in Fig. 4.

In Fig. 4, after the basic feature input, the time suggestion sub-network will expand the temporal receptive field through 3D convolution, down-sample spatial information, and then perform pooling operations to obtain the feature map. Two convolutions are performed on the feature map to predict the offset and score. Next, the feature vectors of the score branch are compared with the ground truth. Candidate

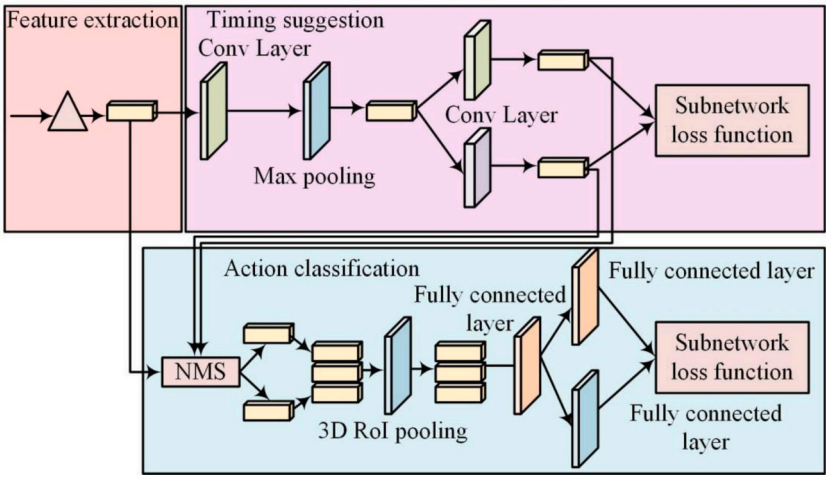


Fig. 3. Structure of the R-C3D.

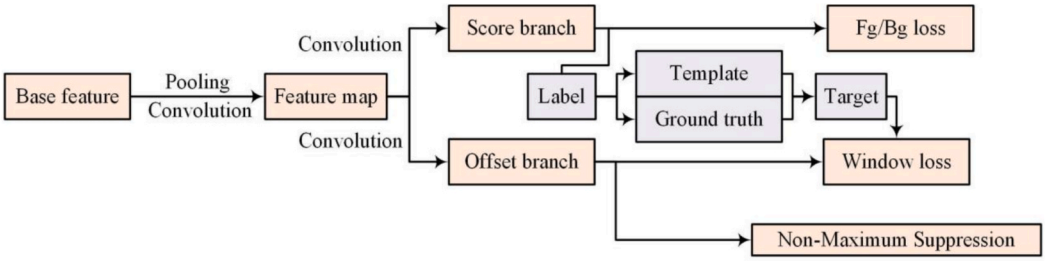


Fig. 4. The time suggestion sub-network structure.

boxes with an Intersection over Union (IoU) greater than 0.7 are used as foreground boxes, while those with an IoU less than 0.3 are used as background boxes. All other candidate boxes are discarded. The feature vectors of the offset branch are compared with the ground truth to obtain the center offset value and length offset value of the standard box. Then, it is used as target offset information. Based on the target offset information, the loss for optimizing the prediction box can be obtained [4,26]. At this point, the standard time-domain candidate boxes are fused with the predicted offset information, and the non maximum suppression algorithm is used to remove duplicate crossed candidate boxes. The action classification sub-network can be used for sign language action classification. The action classification sub-network is displayed in Fig. 5.

From Fig. 5, the predicted box obtained from the time suggestion sub-network is compared with the ground truth of the real sign language information. If the IoU was greater than 0.5, the predicted box was the target box. Otherwise, it was the background box. In the prediction boxes that met the above conditions, several borders were randomly selected in a 1:3 ratio. Based on the information of the selected prediction box, 3D RoI pooling is performed to extract the corresponding size features of the prediction box from the basic features. The fully connected layer is applied to obtain the score and offset information of the predicted box in each category to obtain the classification loss. Simultaneously, the loss of the prediction box is obtained based on the offset of the prediction box and the target offset information [27], Xiang et al. 2021). The classification loss function is shown in Eq. (12).

$$Loss = \frac{1}{N_{cls}}\sum_i L_{cls}\left(a_i, a_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i a_i^* L_{reg}\left(t_i, t_i^*\right) \tag{12}$$

in Eq. (12), N_{cls} represents the batch size. L_{cls} is the video length. a_i signifies the likelihood of a proposal or action prediction. a_i^* represents the true label of the action. λ represents the constant coefficient. N_{reg} represents the number of temporal candidate segments. L_{reg} represents the length of the temporal candidate segment. t_i represents the relative offset of the temporal candidate segment. t_i^* represents the true bounding box for coordinate transformation. The coordinate conversion is shown in Eq. (13).

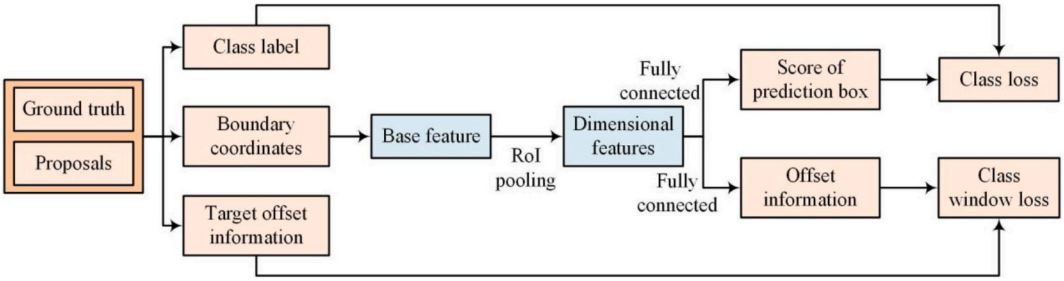


Fig. 5. Structure of the action-classification sub-networks.

$$\begin{cases} x_{c_i} = \left(c_i^* - c_i\right)/l_i \\ y_{l_i} = \log\left(l_i^8/l_i\right) \end{cases} \tag{13}$$

in Eq. (13), (x_{c_i}, y_{l_i}) represents the transformed coordinates. c_i and c_i^* represent the center positions of the temporal candidate segment and the true bounding box. l_i and l_i^* represent the length of the temporal candidate segment and the true bounding box. However, due to the short time interval between sign language actions, it may determine two different actions as one action when generating category labels. The sign language action feature map is shown in Fig. 6.

In Fig. 6, sign language actions have continuity and short intervals, and there is a certain overlap in the feature maps of adjacent actions. When the IoU threshold is set to 0.5, the judgment criteria for actions in the predicted boxes are relatively loose, and some predicted boxes may contain edge features of adjacent actions (i.e. a single candidate segment may cover local features of multiple actions), resulting in the model mistakenly judging candidate segments containing other action segments as target actions, and increasing classification interference. When the IoU threshold is increased to 0.7, the criteria for determining prediction boxes become stricter, and the overlap area between candidate segments and real action features is significantly reduced. Each prediction box only covers the feature map of a single complete action, effectively eliminating edge interference from adjacent actions and making the division of action boundaries more accurate. This visualization directly confirms the necessity of raising the action judgment threshold from 0.5 to 0.7, providing a visual basis for subsequent models to reduce misjudgment rates by increasing the threshold. The IoU is shown in Eq. (14).

$$IoU = \frac{A_i \cap A_j}{A_i \cup A_j} \tag{14}$$

in Eq. (14), A_i and A_j represent two different candidate boxes. The removal of redundant bounding boxes is achieved using a non maximum suppression algorithm. Firstly, candidate boxes with IoU greater than a certain threshold are multiplied by a coefficient to obtain the updated IoU. If the updated IoU of the candidate box is greater than the set threshold, the candidate box is considered redundant and deleted. The specific calculation is shown in Eq. (15).

$$s_i = \begin{cases} s_i & IoU(R, b_i) < N_t \\ s_i(1 - IoU(R, b_i)) & IoU(R, b_i) \geqslant N_t \end{cases} \tag{15}$$

in Eq. (15), s_i represents the window score. R represents the candidate box for the maximum confidence score. b_i represents the remaining candidate boxes. N_t represents the threshold.The sign language recognition process based on the improved R-C3D is as follows: First, the C3D feature extraction network is used to extract the video stream data, then the time range of actions may be generated, the time range is adjusted to make it more accurate, and the feature information corresponding to the time range is selected to complete the classification of actions. The pseudo code for improved R-C3D is shown in Table 2.

4. Dynamic sign language detection results and analysis

This chapter tests the performance of the proposed sign language classification model based on improved R-C3D, which is divided into two parts: sub-network performance testing and overall recognition performance testing. The evaluation indicators include Mean Average Precision (mAP), convergence speed, propensity propagation familiarity, training time, etc. The mAP calculation is shown in Eq. (16).

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{16}$$

in Eq. (16), C represents the number of categories. AP_i represents the average precision of the i th category.

4.1. Comparison of feature extraction sub-nets and time-domain standard boxes

To verify the performance of feature extraction sub-nets and time-domain standard boxes, tests are conducted separately. The open-source dataset used for testing is the THUMOS14 dataset, which contains 20 action categories, each containing 30–400 action fragments. The training set of the THUMOS14 dataset contains 1,010 unedited videos, and the test set contains 1,574 unedited videos. The sign language dataset used for testing is collected through RGB cameras. An RGB camera (model: Logitech C920) with a resolution of 1,920*1,080 pixels is adopted and a fixed frame rate of 30 frames per second is taken to clearly capture motion details. The collection environment is indoor constant lighting (brightness 500–800 lx), with a solid color (white) background to reduce complex background interference. A total of 10 volunteers participate in the recording, including 6 males and 4 females, aged between 22 and 35 years old. All volunteers are proficient in the "National Common Sign Language Commonly Used Vocabulary List" and include three professional sign language translators to standardize movements. Simultaneously, two volunteers with significant body movements are included to increase sample diversity. 15 high-frequency words are selected from the "National Commonly Used Sign Language Vocabulary List", including "hello", "thank you", "goodbye", "yes", "no", "help", "need" et al, covering daily scenarios such as greetings, responses, and expressions of needs. 70–80 action clips for each

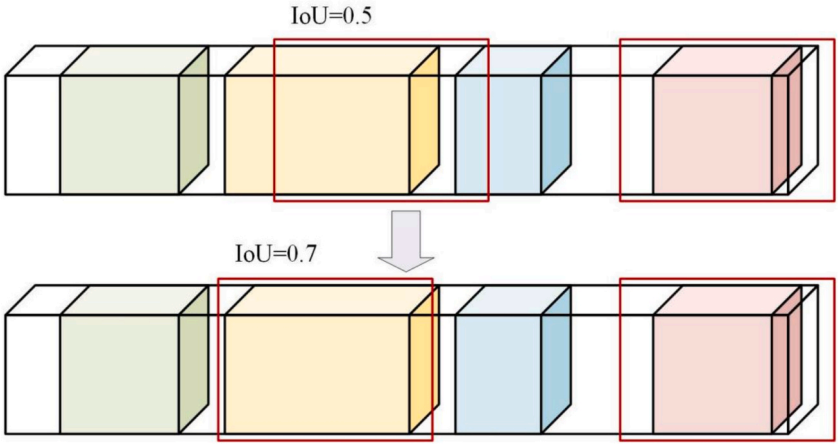


Fig. 6. Feature map of sign language actions.

Table 2
Pseudo code for improved R-C3D.

Pseudo code
#Input: Continuous sign language video clips (frame sequence, shape=(L, H, W, 3), L is the frame rate, H/W is the image size)
#Output: Action category labels and corresponding time boundaries (start_frame, end_frame)
1. Feature extraction sub-network (improved C3D: depthwise separable convolution + inverse residual structure)
def modified_C3D(frames):
#Initial 3D convolution (3 × 3 × 3)
x = Conv3D(kernel=(3,3,3), filters = 32, stride=(1,1,1))(frames)
X = MaxPool3D (kernel=(2,2,2)) (x) # Pooling dimension reduction
#Inverted residual module (depthwise separable convolution + skip connection)
def inverted_residual(x, expansion, out_filters):
#Point by point convolution to increase dimensionality (expansion factor = 6)
x = Conv3D(kernel=(1,1,1), filters = x.shape[-1]*expansion)(x)
x = BatchNorm()(x)
X = ReLU6() (x) # Limit activation value range
#Deep convolution (channel wise convolution)
x = DepthwiseConv3D(kernel=(3,3,3), stride=(1,1,1))(x)
x = BatchNorm()(x)
x = ReLU6()(x)
#Point by point convolution for dimensionality reduction
x = Conv3D(kernel=(1,1,1), filters = out_filters)(x)
Return x + x_rev # jump connection (residual)
#Stacking 4 inverted residual modules (see Table 1 for parameters)
x = inverted_residual(x, expansion = 6, out_filters = 64)
x = inverted_residual(x, expansion = 6, out_filters = 128)
x = inverted_residual(x, expansion = 6, out_filters = 256)
x = inverted_residual(x, expansion = 6, out_filters = 256)
X = AvgPool3D() (x) # Global average pooling
Return x # Output spatiotemporal feature map (shape=(T, H ', W', C), where T is the temporal dimension)
2. Time suggestion sub-network (reduce pre-selection box + NMS)
def temporal_proposal_net(feature_map):
#Generate pre selection boxes (reduced to 8 scales, adapted to 30–60 frames of sign language actions)
Anchors = generate_anchors (scales= [2,4,6,8,10,12,14,16]# Unit: frames
#Prediction box offset and foreground/background score
offsets, scores = Conv3D(kernel=(1,1,1), filters = 2*len(anchors))(feature_map)
#Filter foreground boxes (IoU ≥ 0.7 for foreground,<0.3 for background)
positive_anchors = [anchor for i, anchor in enumerate(anchors) if scores[i] ≥ 0.7]
#Bounding box regression (corrected prediction box position)
refined_boxes = apply_offsets(positive_anchors, offsets)
#Non Maximum Suppression (NMS) removes duplicate boxes
nms_boxes = NMS(refined_boxes, iou_threshold = 0.5)
Return nms_boxes # Output candidate action time boxes (start, end)
3. Action classification sub-network (increasing decision threshold)
def action_classification_net(feature_map, proposals):
#3D RoI pooling (extracting features corresponding to candidate boxes)
roi_features = RoIAlign3D(feature_map, proposals, output_size=(1,1,1))
#Classification and bounding box fine-tuning
Cls_scores, bbox_preads = Dense (units = num_classes + 4) (roi_features) # Category + bounding box offset
#Screen high confidence predictions (threshold = 0.7)
final_preds = [
(class_id, bbox)
for cls_score, bbox in zip(cls_scores, bbox_preds)
if max(cls_score) ≥ 0.7
]
Return final_rides # Output (action category ID, time boundary)
#Overall Model Process
def modified_R_C3D(video_frames):
#Step 1: Extract spatiotemporal features
features = modified_C3D(video_frames)
#Step 2: Generate candidate action time frames
proposals = temporal_proposal_net(features)
#Step 3: Action classification and bounding box optimization
results = action_classification_net(features, proposals)
return results
#Training process (loss function)
def train_step(video_frames, ground_truth):
#Forward propagation
preds = modified_R_C3D(video_frames)
#Calculate loss (classification loss + bounding box loss)
cls_loss = CrossEntropyLoss(preds.class_ids, ground_truth.class_ids)
bbox_loss = SmoothL1Loss(preds.bboxes, ground_truth.bboxes)
Total loss = cls_loss + 1.0 * bbox-loss # loss weighting
#Backpropagation optimization
optimizer.zero_grad()
total_loss.backward()
optimizer.step()
return total_loss

vocabulary are recorded, totaling over 700 clips (with an average of 70 samples per vocabulary). Each segment corresponds to a complete dynamic sign language action, lasting 1–2 s (corresponding to 30–60 frames of video).

For the collected sign language dataset, the video is segmented using a "keyframe difference + manual verification" approach. Firstly, the pixel variation of adjacent frames is calculated using the inter frame difference algorithm. When the variation continuously exceeds the threshold (set as 20 %, which is the pixel difference at the beginning/end of the action), it is marked as a candidate boundary. Subsequently, two sign language experts verify the candidate boundaries and adjust the boundary frames based on the semantic integrity of sign language actions (such as the complete process of raising, waving, and lowering the hand with "hello"), ultimately determining the start and end frames of each action segment. The research is conducted on using the open-source video annotation tool Label Studio to annotate videos. The GPU used in the experiment is GTX 1080Ti 11 GB, with an Intel(R) Core (TM)i79700K 3.6 GHz CPU. The model framework is built using Python + Python, and the running environment is managed through Anaconda 3. The testing of feature extraction sub-nets is performed on the THUMOS14 dataset, while the comparison of time-domain standard boxes is conducted on the sign language dataset. In the experiment, the learning rate of the algorithm is 0.01, the batch size is 16, the learning rate decay coefficient is 0.1, and the optimizer is Adam. The comparison models are R-C3D, Net-13, and Net-27, where the structures of Net-13 and Net-27 are shown in Table 3.

According to Table 3, compared to the C3D network, the Net-13 network and Net-27 network have deeper network depths, and their feature channel numbers are only half of the original (256 channels, while the output channel number of C3D is 512).The test results of the feature extraction sub-net are shown in Fig. 7.

In Fig. 7(a), the scale of the improved C3D network significantly decreased compared with the original C3D, with a size of only 288 MB, but slightly larger than Net-13 and Net-27. In terms of training time, the time required for the improved C3D training was only 3.2 h, less than Net-27, but slightly longer than Net-13. From Fig. 7(b), in terms of forward propagation speed, the propagation of the improved C3D only was 172 ms, second only to Net-13. In addition, the improved C3D had a mAP of 44.2 %, which was much higher than that of other networks. The above results indicate that the comprehensive performance of the optimized C3D network is superior to other networks.

Fig. 8 displays the loss curve and mAP curve of the feature extraction sub-net. From Fig. 8(a), after training 10,000 steps, the loss value of C3D gradually decreased slowly. After training 10,000 steps, the loss values of Net-13 and Net-27 still maintained a relatively fast decline rate, while the loss value of the improved C3D decreased faster than both Net-13 and Net-27. In Fig. 8(b), the mAP of each network gradually decreased with the increase of IoU. When the IoU was 0.5, the mAP of C3D, Net-13, Net-27, and improved C3D were 38.5 %, 35.3 %, 40.1 %, and 44.2 %, respectively. When the IoU was 0.7, the mAP of each network was 18.9 %, 16.1 %, 20.0 %, and 25.2 %, respectively. The mAP of improved C3D is much higher than other networks, and it is less affected by changes in IoU.

Table 3
The structures of Net-13 and Net-27.

Layer	Net-13	Net-27
Conv 1	7*7*7, 64, BatchNormal, ReLU	7*7*7, 64, BatchNormal, ReLU
Maxpool	3*3*3	3*3*3
Basicblock 1	{3*3*3, 64, BatchNormal, ReLU, 3*3*3, 64, BatchNormal}*2	{3*3*3, 64, BatchNormal, ReLU, 3*3*3, 64, BatchNormal}*3
Basicblock 2	{3*3*3, 128, BatchNormal, ReLU, 3*3*3, 128, BatchNormal}*2	{3*3*3, 128, BatchNormal, ReLU, 3*3*3, 128, BatchNormal}*4
Basicblock 3	{3*3*3, 256, BatchNormal, ReLU, 3*3*3, 256, BatchNormal}*2	{3*3*3, 256, BatchNormal, ReLU, 3*3*3, 256, BatchNormal}*6

Fig. 9 shows the comparison results of different long and short time domain standard boxes. In Fig. 9(a), under the same other conditions, compared with the original pre-selection box, the reduced pre-selection box had a significant increase in mAP, from 41.6 % to 44.5 %. This is because after the pre-selection box is reduced, the foreground background interception is more accurate and closer to the ground truth. From Fig. 9 (b), the loss value of the time recommendation sub-network for the standard pre-selected box gradually converged after 20,000 iterations, with a loss value of approximately 0.50. The time suggestion sub-network loss value of the reduced pre-selection box gradually converged after 17,000 iterations, and its loss value was about 0.46. The above results indicate that shortening the pre-selection box can effectively improve the mAP and convergence speed.

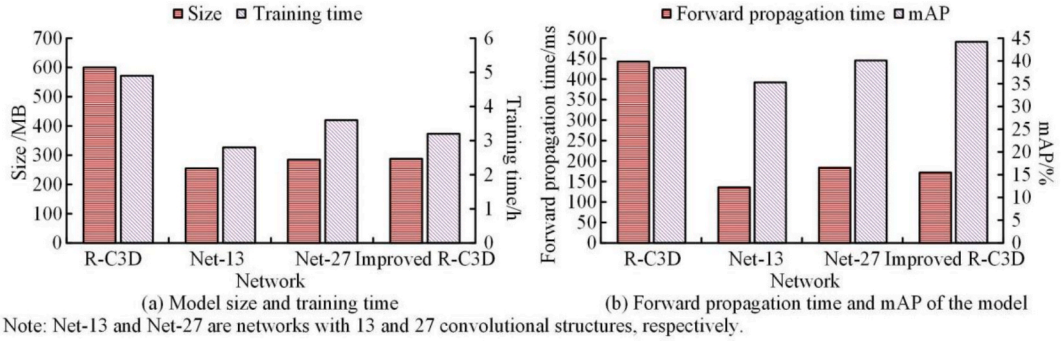
Fig. 10 shows the foreground/background prediction box and classification loss curve. From Fig. 10(a), there was little change in the foreground/background classification loss curve before and after reducing the pre-selection box, only a slight improvement in the convergence speed of the loss curve after the reduction of the pre-selection box. This indicates that shortening the size of the pre-selection box slightly improves the convergence efficiency of the model for foreground/background classification, but the overall classification loss is not significantly different, indicating that the influence of pre-selection box size on the basic classification logic is limited. In Fig. 10(b), the loss value of the foreground/background prediction box of the standard pre-selection box gradually converged after 40,000 iterations, with a loss value of around 0.30. Although the loss value of the foreground/background prediction box that reduced the pre-selection box also converged after approximately 40,000 iterations, its loss value was only about 0.18. It can be seen that shortening the pre-selection box can effectively reduce the loss value of the foreground/background prediction box. This is because the size of the original pre-selection box is relatively large, and the matching degree with the actual duration of sign language actions is low, which easily contains too much irrelevant background information, resulting in a large deviation between the predicted box and the real action boundary. The shortened pre-selection box is more in line with the actual duration of sign language actions, reducing redundant information interference and enabling the model to more accurately locate action boundaries, thereby reducing the loss value of the prediction box.

Fig. 11 shows the action classification and prediction box loss curve. From Fig. 11(a), the action classification loss curve before and after reducing the pre-selection box was basically the same. Only in the early stages of iteration, the loss value of the standard pre-selection box was smaller than that of the reduced pre-selection box. In Fig. 11(b), after 10,000 iterations, the loss value of the action prediction box for reducing the pre-selection box was always smaller than that of the standard pre-selection box, with values of 0.21 and 0.19, respectively. The above results indicate that reducing the size of the pre-selection box effectively reduces the loss value of the action prediction box, making it more accurate in predicting the start time of the action.

4.2. Comparison of action decision thresholds and continuous sign language recognition

Due to the short time interval between consecutive sign language actions, the prediction box may misjudge other actions as target actions. Therefore, to understand the impact of action decision thresholds on the accuracy of sign language recognition, a study is conducted on action classification sub-networks with decision thresholds of 0.5 and 0.7, using the sign language dataset as the test dataset. The overall recognition performance is also conducted on the sign language dataset. The loss curve and mAP of the action classification sub-network are shown in Fig. 12.

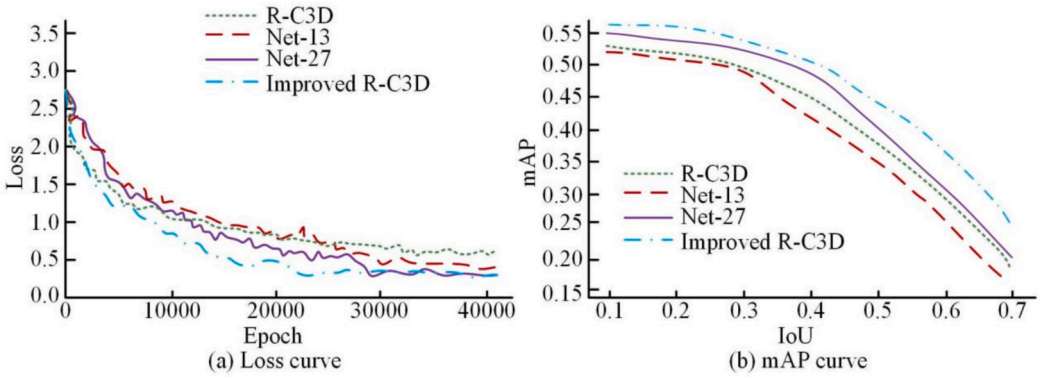
In Fig. 12(a), the loss value of the action classification sub-network with a judgment threshold of 0.5 gradually converged after 15,000



(a) Model size and training time (b) Forward propagation time and mAP of the model

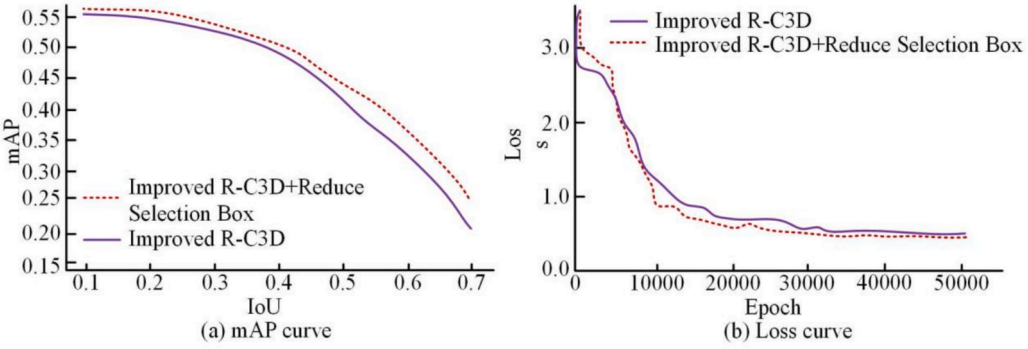
Note: Net-13 and Net-27 are networks with 13 and 27 convolutional structures, respectively.

Fig. 7. Test results of the feature extraction sub-net.



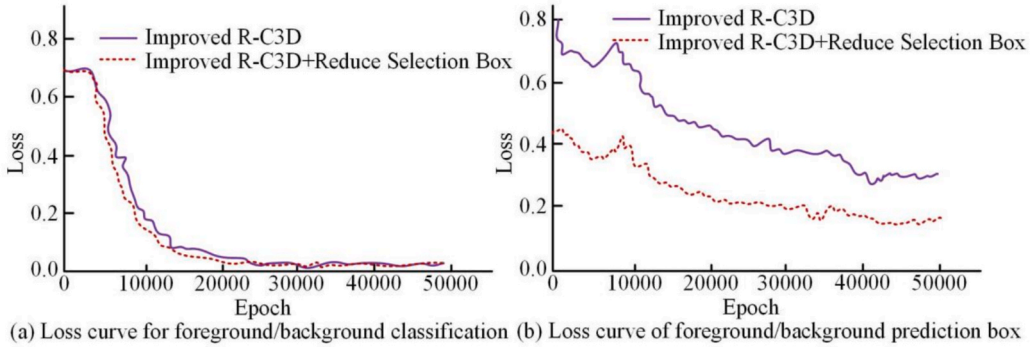
(a) Loss curve (b) mAP curve

Fig. 8. Loss curves and mAP curves of the feature extraction sub-net.



(a) mAP curve (b) Loss curve

Fig. 9. Comparison results of standard boxes for different time lengths.



(a) Loss curve for foreground/background classification (b) Loss curve of foreground/background prediction box

Fig. 10. Foreground / background prediction box and classification loss curves.

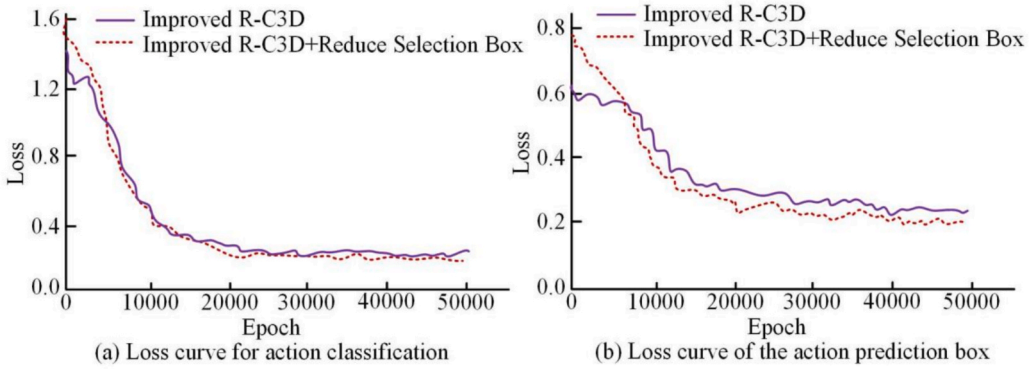


Fig. 11. Action classification and predicted box loss curves.

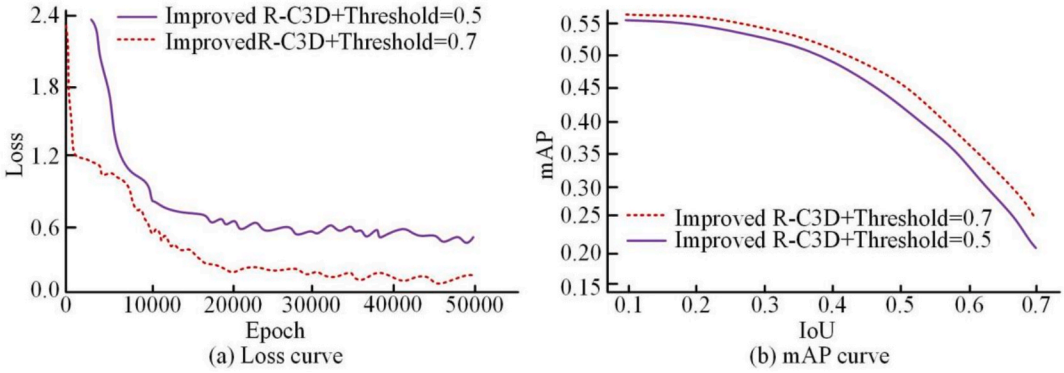


Fig. 12. Loss curves and mAP for action classification sub-networks.

iterations, and its loss value was 0.58. The loss value convergence speed with a judgment threshold of 0.7 was not significantly different from that of the sub-net with a judgment threshold of 0.5, but its loss value was only 0.17. In Fig. 12(b), when the action judgment threshold was 0.5, the mAP with IoU of 0.5 was 43.4 %. When the action judgment threshold was 0.7, the mAP with IoU of 0.5 was 45.5 %, an increase of 2.1 %. The above results indicate that when the action judgment threshold is raised to 0.7, the convergence speed and mAP of the action classification sub-net are significantly improved.This is because the prediction box after the high IoU threshold screening is very close to the ground truth, which makes the subsequent action features obtained more pure, so that the action features can be accurately extracted.

Fig. 13 shows the loss curve of action classification and prediction boxes. In Fig. 13(a), the convergence speed of the action classification loss with a judgment threshold of 0.5 was slightly faster than that of the network with a decision threshold of 0.7. However, the action classification loss with a judgment threshold of 0.7 was only 0.03, which was much smaller than the network with a judgment threshold of 0.5. In Fig. 13(b), the loss value of the action prediction box with a judgment threshold of 0.5 gradually converged after 35,000 iterations, with a loss value of approximately 0.22. The action prediction box with a judgment threshold of 0.7 was very close to the ground truth, and the loss value remained around 0.01. The above results indicate that increasing the action judgment threshold to 0.7 can effectively improve the loss value of action classification and prediction boxes.

Fig. 14 shows the loss curves of different sign language recognition schemes. In Fig. 14, R-C3D, improved R-C3D + Threshold = 0.5, and improved R-C3D + Threshold = 0.7 gradually converged after 30,000, 20,000, and 25,000 iterations, respectively, with loss values of approximately 0.51, 0.50, and 0.21. The loss values of improved R-C3D

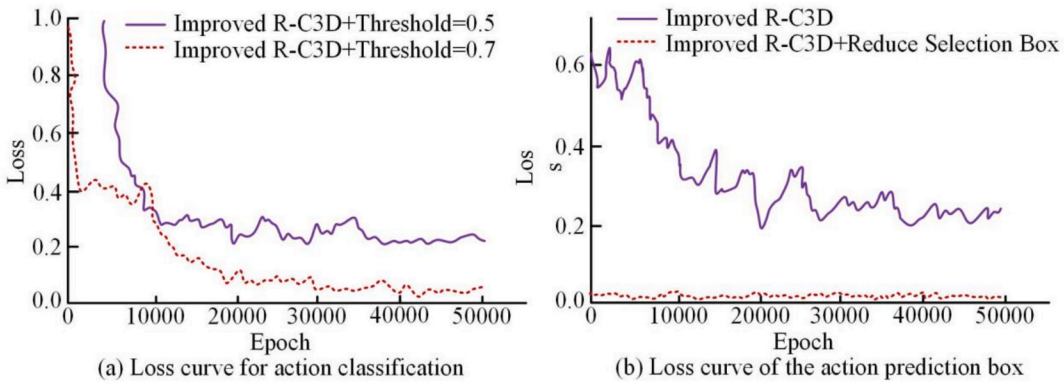


Fig. 13. Loss curves of action classification and prediction box.

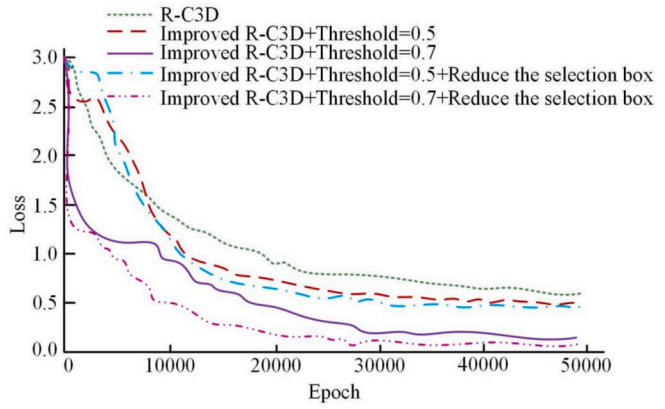


Fig. 14. Loss curves for different sign language recognition schemes.

+ Threshold = 0.5 + reduced pre-selection boxes and improved R-C3D + Threshold = 0.7 + reduced pre-selection boxes gradually converged after 25,000 and 20,000 iterations, respectively, with loss values of 0.49 and 0.15. The above results indicate that the improved R-C3D + Threshold = 0.7 + reduced pre-selection box sign language recognition scheme proposed in the study has a smaller loss value and faster convergence speed.

Fig. 15 shows the recognition time and mAP of different sign language recognition schemes. According to Fig. 15(a), the average recognition speeds of R-C3D, improved R-C3D + Threshold = 0.5, and improved R-C3D + Threshold = 0.7 were 440 ms, 193 ms, and 189 ms, respectively. The average recognition speeds of improved R-C3D + Threshold = 0.5 + reduced pre-selection boxes and improved R-C3D + Threshold = 0.7 + reduced pre-selection boxes were 187 ms and 183 ms, respectively. According to Fig. 15(b), the mAP of R-C3D, improved R-C3D + Threshold = 0.5, and improved R-C3D + Threshold = 0.7 at an IoU of 0.5 were 38.8 %, 40.1 %, and 42.5 %, respectively. The mAP of improved R-C3D + Threshold = 0.5 + reduced pre-selection boxes and improved R-C3D + Threshold = 0.7 + reduced pre-selection boxes at an IoU of 0.5 were 41.2 % and 44.6 %, respectively. The above results indicate that the improved R-C3D + Threshold = 0.7 + reduced pre-selection box sign language recognition scheme proposed in the study not only has a faster recognition speed than other schemes, but also has a much higher mAP than that of other schemes. To further investigate the performance of the proposed sign language recognition algorithm based on improved R-C3D, it was compared with the methods in Rajalakshmi et al [28] and Rajalakshmi et al [29]. The accuracy and recall of sign language recognition for each algorithm are shown in Fig. 16.

According to Fig. 16(a), the highest accuracy for sign language recognition in Rajalakshmi et al [28] and Rajalakshmi et al [29] was 92.4 % and 93.8 %, respectively. Compared with the methods in Rajalakshmi et al [28] and Rajalakshmi et al [29], the improved R-C3D had a higher accuracy in sign language recognition, with a minimum recognition accuracy of 96.7 % and a maximum recognition accuracy of 97.7 %. According to Fig. 16(b), the highest recall rates for sign language recognition in Rajalakshmi et al [28] and Rajalakshmi et al [29] were 89.7 % and 90.4 %, respectively, while the improved R-C3D had the highest recall rate of 92.8 %, which was higher than that of other algorithms. The above results indicate that the improved R-C3D can accurately recognize sign language. The memory consumption of each algorithm is shown in Table 4.

According to Table 4, compared to other algorithms, the proposed method had a smaller memory footprint of only 288 MB. Compared to other methods, this decreased by at least 11.1 %. The results of the ablation experiment are shown in Table 5.

According to Table 5, among all modules, the depthwise separable convolution had the most significant impact on algorithm performance. Compared with the original model, by introducing depthwise separable convolution, the mAP increased to 41.5 %, with an increase of 2.7 %.

5. Conclusion

Sign language uses gestures to measure actions, simulating images or syllables on the basis of changes in gestures to form certain meanings or words. It is the primary communication tool for deaf and mute people. However, due to the low popularity of sign language in society, communication between deaf and mute individuals and the normal population is more difficult. Therefore, to facilitate communication, a continuous sign language recognition model based on improved R-C3D was proposed, which used the improved C3D network as the feature extraction sub-network. The mAP of the improved C3D network was 44.2 %, while the mAP of C3D, Net-13, and Net-27 were 38.5 %, 35.3 %, and 40.1 %, respectively. The mAP of improved C3D was higher than that of other networks. In addition, after training 10,000 steps, the decrease in loss value of C3D gradually slowed down. After training 10,000 steps, the loss values of Net-13 and Net-27 still maintained a relatively fast decline rate, while the loss value of the improved C3D decreased faster than that of both Net-13 and Net-27. After reducing the pre-selection box, the mAP of the time suggestion sub-network increased significantly, from 41.6 % to 44.5 %. The convergence speed significantly accelerated, and the loss value also decreased from 0.5 to 0.46. When the action judgment threshold increased from 0.5 to 0.7, the convergence speed of the loss value was not significantly different from the sub-net with a judgment threshold of 0.5, but its loss value decreased from 0.58 to 0.17. The entire improved R-C3D network gradually converged after 20,000 iterations, with a loss value of only 0.15, a sign language recognition speed of 183 ms, and an mAP of 44.6 %, all of which were superior to those of other sign language recognition schemes. E. Hassan et al. [15] proposed the Sign Nevestro Densenet Attention (SNDA) method to address the American Sign Language recognition. This method used Nadam optimizer to achieve faster

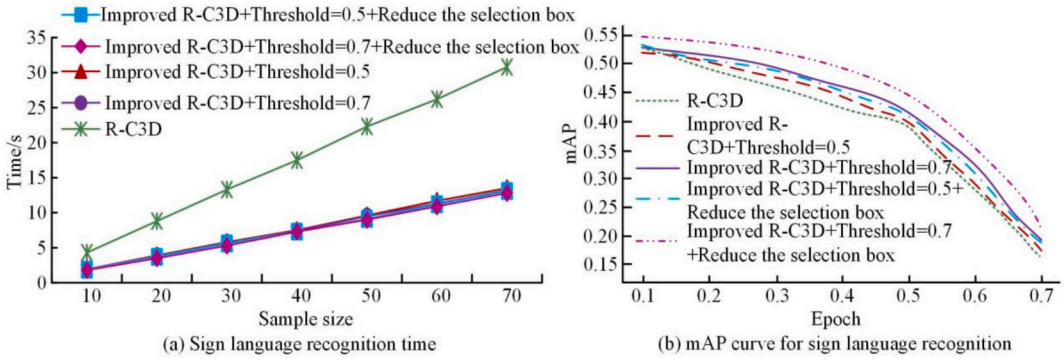


Fig. 15. Identification times and mAP for different sign language recognition schemes.

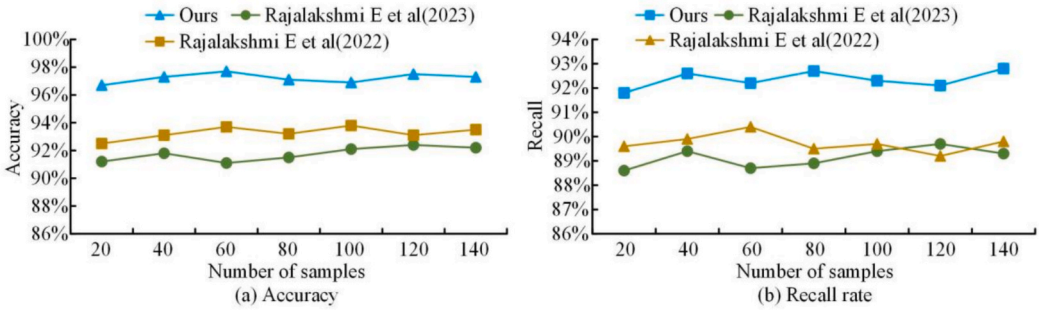


Fig. 16. Sign language recognition accuracy and recall rate of each algorithm.

Table 4
Memory consumption of each algorithm.

Model	Hardware environment	Memory usage/MB
Rajalakshmi E et al. (2023)	GTX 1080Ti 11 GB GPU, Intel (R) Core (TM) i7-9700 K 3.6 GHz CPU	375
Rajalakshmi E et al. (2022)	GTX 1080Ti 11 GB GPU, Intel (R) Core (TM) i7-9700 K 3.6 GHz CPU	324
Ours	GTX 1080Ti 11 GB GPU, Intel (R) Core (TM) i7-9700 K 3.6 GHz CPU	288

Table 5
Results of ablation experiment.

Model	Depthwise separable convolution	Inverse residual	Reduce the pre-selection box	mAP(IoU = 0.5)/%
1	×	×	×	38.8
2	√	×	×	41.5
3	×	√	×	40.4
4	×	×	√	41.3
5	√	√	×	43.6
6	√	×	√	42.7
7	×	√	√	42.2
8	√	√	√	44.6

training convergence and integrated attention mechanism to enhance model performance. The results showed that SNDA achieved an accuracy of 99.76 %, perfect sensitivity, high specificity and precision, verifying its effectiveness in American Sign Language recognition and potential to promote inclusivity in deaf and hard of hearing communities. He [30] proposed an HMI method based on ultrasound signals to address the application of gesture recognition human–machine interface (HMI) in prosthetic control. This method achieved gesture recognition through the principle of ultrasound, and provided a detailed introduction to the workflow, performance evaluation, robustness of HMI, as well as the latest developments in wearable ultrasound systems. It also summarized its various applications in gesture recognition. The results show that ultrasound-based HMI has significant research progress and broad application prospects in the field of gesture recognition, but still faces some research challenges. Future research directions are worth further exploration. Compared to the above methods, the proposed method has faster recognition speed, higher accuracy, and smaller memory footprint. The above results indicate that the sign language recognition scheme based on improved R-C3D has fast recognition speed and high accuracy. In summary, the proposed method of "improved R-C3D + depthwise separable convolution + inverse residual structure + optimized pre-selection boxes and decision thresholds" significantly improves existing technologies by addressing the three core issues of "spatiotemporal feature coupling extraction", "computational cost control", and "action boundary accuracy" in dynamic sign language recognition, especially compared to models that rely on Transformers or GCNs, demonstrating unique advantages. Compared to the Transformer model, the improved R-C3D convolution kernel slides directly in the "height width time" 3D space to achieve native coupling extraction of spatiotemporal features. Compared to GCN, the improved R-C3D convolution directly extracts temporal dynamic features through sliding windows of consecutive frames, and utilizes multi-channel feature fusion through depthwise separable convolution to suppress background noise and local occlusion effects. Although the proposed sign language recognition scheme based on R-C3D has good recognition accuracy, the model size and training time are still relatively high. Meanwhile, due to the strong targeting of the model, its transfer learning potential is compromised. Therefore, future research considers introducing methods such as pruning, quantization, and knowledge extraction to reduce the memory usage of the model to below 200 MB and compress the single recognition time to within 100 ms. Simultaneously, training strategies can be optimized (such as using mixed precision training and distributed training) to shorten training time to less than 2 h, meeting the requirements of edge device deployment and fast iteration.

6. Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article.

CRediT authorship contribution statement

Haofei Chen: Writing – original draft, Formal analysis, Data curation, Conceptualization. **Chang'an Di:** Writing – review & editing, Validation, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Guo L, Lu Z, Yao L. Human-machine interaction sensing technology based on hand gesture recognition: a review. IEEE Trans Hum-Mach Syst 2021;51(4):300–9. https://doi.org/10.1109/THMS.2021.3086003.
[2] Moin A, Zhou A, Rahimi A, Menon A, Benatti S, Alexandrov G, et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. Nat Electron 2020;4(1):54–63. https://doi.org/10.1038/s41928-020-00510-8.
[3] Qi W, Ovur S-E, Li Z, Marzullo A, Song R. Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network. IEEE Rob Autom Lett 2021;6(3):6039–45. https://doi.org/10.1109/LRA.2021.3089999.
[4] Liu H, Zhou A, Dong Z, Sun Y, Zhang J, Liu L, et al. M-gesture: Person-independent real-time in-air gesture recognition using commodity millimeter wave radar. IEEE Internet Things 2022;9(5):3397–415. https://doi.org/10.1109/JIOT.2021.3098338.
[5] Wong W-K, Juwono F-H, Khoo B-T-T. Multi-features capacitive hand gesture recognition sensor: a machine learning approach. IEEE Sens J 2021;21(6):8441–50. https://doi.org/10.1109/JSEN.2021.3049273.

[6] Saboo S, Singha J, Laskar R-H. Self co-articulation removal and hybrid classifier-feature combination for dynamic hand gesture recognition. Multimed Tools Appl 2022;82(4):6033–52. https://doi.org/10.1007/s11042-022-13571-y.
[7] Hu M, He P, Zhao W, Zeng X, He J, Chen Y, et al. Machine learning-enabled intelligent gesture recognition and communication system using printed strain sensors. ACS Appl Mater Interfaces 2023;15(14):51360–9. https://doi.org/10.1021/acsami.3c10846.
[8] Khattak AS, Zain ABM, Hassan RB, Nazar F, Haris M, Ahmed BA. Hand gesture recognition with deep residual network using semg signal. Biomed Eng/Biomedizinische Technik 2024;69(3):275–91. https://doi.org/10.1515/bmt-2023-0208.
[9] Chen C, Yu Y, Sheng X, Meng J, Zhu X. Real-time hand gesture recognition by decoding motor unit discharges across multiple motor tasks from surface electromyography. IEEE Trans Biomed Eng 2023;70(7):2058–68. https://doi.org/10.1109/TBME.2023.3234642.
[10] Lu Y, Le V-L, Kim T-H. A 184-μw error-tolerant real-time hand gesture recognition system with hybrid tiny classifiers utilizing edge CNN. IEEE J Solid State Circuits 2022;589(2):530–42. https://doi.org/10.1109/JSSC.2022.3179601.
[11] Tao Z, Shuguo P, Wang G, Yingchun S. Learning modal and spatial features with lightweight 3d convolution for rgb guided depth completion. IEEE Trans Consum Electron 2021;67(3):195–201. https://doi.org/10.1109/TCE.2021.3095378.
[12] Wu C, Yi X, Wang W, You L, Huang Q, Gao X, et al. Learning to localize: a 3d cnn approach to user positioning in massive mimo-of dm systems. IEEE Trans Wirel Commun 2021;20(7):4556–70. https://doi.org/10.1109/TWC.2021.3060482.
[13] Kozlovskii I, Popov P. Protein-peptide binding site detection using 3d convolutional neural networks. J Chem Inf Model 2021;61(8):3814–23. https://doi.org/10.1021/acs.jcim.1c00475.
[14] Hassan E, Elbedwehy S, Shams MY, Abd El-Hafeez T, El-Rashidy N. Optimizing poultry audio signal classification with deep learning and burn layer fusion. J Big Data 2024;11(1). https://doi.org/10.1186/s40537-024-00985-8.
[15] Hassan E, Saber A, El-Hafeez TA, Medhat T, Shams MY. Enhanced dysarthria detection in cerebral palsy and ALS patients using WaveNet and CNN-BiLSTM models: a comparative study with model interpretability. Biomed Signal Process Control 2025;110. https://doi.org/10.1016/j.bspc.2025.108128.
[16] Bhosle K, Musande V. Evaluation of deep learning CNN Model for recognition of devanagari digit. Artificial Intelligence and Appl 2023;1(2):114–8. https://doi.org/10.47852/bonviewAIA3202441.
[17] Preethi P, Mamatha H-R. Region-based convolutional neural network for segmenting text in epigraphical images. Artificial Intelligence and Appl 2023;1(2):119–27. https://doi.org/10.47852/bonviewAIA2202293.
[18] Wu Y, Kong D, Wang S, Li J, Yin B. Hpgcn: hierarchical poselet-guided graph convolutional network for 3d pose estimation. Neurocomputing 2022;487(28):243–56. https://doi.org/10.1016/j.neucom.2021.11.007.
[19] Kumar D, Kumar D-S. A spectral–spatial 3d-convolutional capsule network for hyperspectral image classification with limited training samples. Int J Inf Technol 2022;15(1):379–91. https://doi.org/10.1007/s41870-022-01075-9.
[20] Qiu K, Chen W, Shen J, Zhou H. A novel 3d convolutional neural network model with supervised spectral regression for recognition of hyperspectral images of colored wool fiber. Color Res Appl 2022;47(5):1105–17. https://doi.org/10.1002/col.2278.
[21] Yaosen C, Guo B, Shen Y, Wang W, Lu W, Suo X. Capsule boundary network with 3d convolutional dynamic routing for temporal action detection. IEEE Trans Circuits Syst Video Technol 2022;32(5):2962–75. https://doi.org/10.1109/TCSVT.2021.3104226.
[22] Zhu K, Lu W, Liu J, Luo X, Zhao X. A lightweight 3d convolutional neural network for deepfake detection. Int J Intell Syst 2021;36(9):4990–5004. https://doi.org/10.1002/int.22499.
[23] Jung S, Jeoung J, Kang H, Hong T. 3d convolutional neural network-based one-stage model for real-time action detection in video of construction equipment. Comput Aided Civ Inf Eng 2021;37(1):126–43. https://doi.org/10.1111/mice.12695.
[24] Masoudi B, Daneshvar S, Razavi S-N. Multi-modal neuroimaging feature fusion via 3d convolutional neural network architecture for schizophrenia diagnosis. Intell Data Anal 2021;25(3):527–40. https://doi.org/10.3233/IDA-205113.
[25] Hernandez L-J-R, Dominguez H-D-J-O, Villegas O-O-V, Sanchez V-G-C, Gonzalez J-P, Azuela J-H-S. Residual 3d convolutional neural network to enhance sinograms from small-animal positron emission tomography images. Pattern Recogn Lett 2023;172(8):267–73. https://doi.org/10.1016/j.patrec.2023.05.005.
[26] He L, Ding B, Wang H, Zhang T. An optimal 3d convolutional neural network based lipreading method. IET Image Proc 2021;16(1):113–22. https://doi.org/10.1049/ipr2.12337.
[27] Liu Y, Mei Q, Gan X, Zhu Y, Wang Y. Design of action detection system in wrestling match video based on 3d convolutional neural network. IJWMC 2022;22(1):29–37. https://doi.org/10.1504/ijwmc.2022.122483.
[28] Huang Z, Lin Z, Gong Z, Chen Y, Tang Y. A two-phase knowledge distillation model for graph convolutional network-based recommendation. Int J Intelligent Sys 2022;37(9):5902–23. https://doi.org/10.1002/int.22819.
[29] Xiang, X., R. Abdein,M. Zhai, and N. Lv.2021. 3d point convolutional network for dense scene flow estimation. *Neural Processing Letters.* 54(2) 1155-1173. doi: 10.1007/s11063-021-10673-w.
[30] Rajalakshmi E, Elakkiya R, Subramaniyaswamy V, et al. Multi-semantic discriminative feature learning for sign gesture recognition using hybrid deep neural architecture. IEEE Access 2023;11:2226–38. https://doi.org/10.1109/ACCESS.2022.3233671.