IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 1 October 2023, accepted 31 October 2023, date of publication 7 November 2023, date of current version 14 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330863

RESEARCH ARTICLE

Deep Learning-Based Standard Sign Language Discrimination

MENGLIN ZHANG, SHUYING YANG, AND MIN ZHAO
School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin 300384, China

Corresponding author: Shuying Yang (yangshuying@email.tjut.edu.cn)

This work was supported in part by the Key Cultivation Project of Teaching Achievement Award of the Tianjin Academy of Educational Science and Planning, in 2019, under Grant PYGJ-015; in part by the 2019 Tianjin Virtual Simulation Experiment Teaching Project (Jin Jiao Zheng Ban [2019] No. 69); and in part by the School Level Key Teaching Fund Project of Tianjin University of Technology, in 2020, under Grant ZD20-04.

ABSTRACT General sign language recognition models are only designed for recognizing categories, i.e., such models do not discriminate standard and nonstandard sign language actions made by learners. It is inadequate to use in a sign language education software. To address this issue, this paper proposed a sign language category and standardization correctness discrimination model for sign language education. The proposed model is implemented with a hand detection and standard sign language discrimination method. For hand detection, the proposed method utilizes flow-guided features and acquires relevant proposals using stable and flow key frame detections. This model can resolve the inconsistency between the forward optical flow and the box center point offset. In addition, the proposed method employs an encoder-decoder model structure for sign language correctness discrimination. The encoder model combines 3D convolution and 2D deformable convolution results with residual structures, and it implements a sequence attention mechanism. A Sign Language Correctness Discrimination dataset (SLCD dataset) was also constructed in this study. In this dataset, each sign language video has two recognition labels, i.e., sign language category and standardization category. The semi-supervised learning method was employed to generate pseudo hand position labels. The hand detection model was getting sufficiently high hand detection result. The sign language correctness discrimination model was tested with hand patches or full images. SLCD dataset is available at https://dx.doi.org/10.21227/p9sn-dz70.

INDEX TERMS Continuous sign language recognition, encoder-decoder, tubelet, video object detection, 3D convolution.

I. INTRODUCTION

Compared to gesture recognition, the sign language recognition task is more complex and diverse. Sign language translation based on sign language recognition [1], [2] facilitates communication between the hearing impaired and those with functional hearing. Sign language recognition also promotes the intelligence of human-machine interaction in sign language education [3]. Previously, teaching sign language primarily relied on manual techniques and video content. However, with manual techniques, the availability of demonstrations is limited due to manpower issues. In addition, video teaching does not provide efficient and effective feedback about incorrect sign language actions. Thus, there is an urgent need for sign language teaching software that utilize a deep learning method to identify the correctness of the signs made by sign language students.

Computer vision methods with image and video information are commonly used for sign language recognition. Sign language recognition based on computer vision primarily involves two problems, i.e., the isolated sign language recognition and continuous sign language recognition tasks. Isolated sign language recognition can recognize a single sign language category. For example, Wang et al. [4] employed

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

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

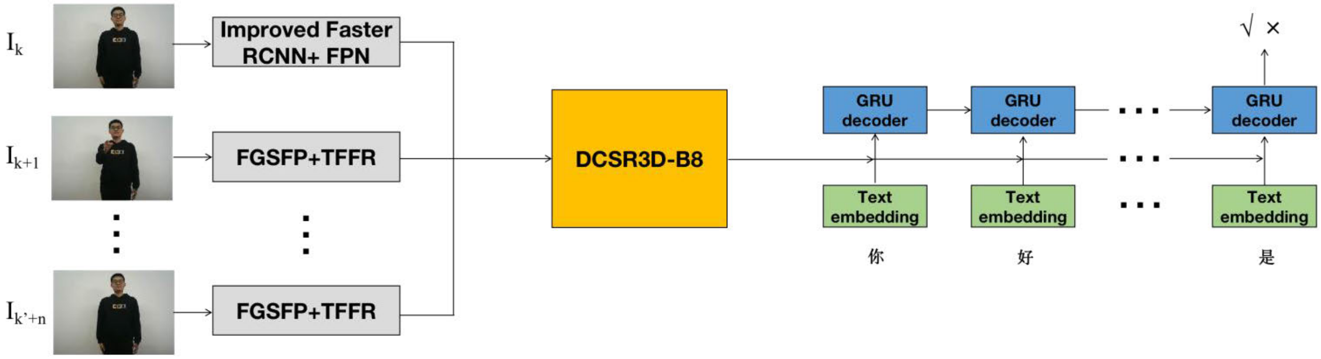

FIGURE 1. **Overall structure of the proposed hand detection and continuous sign language correctness discrimination model. The improved Faster RCNN is employed for hand detection in key frames, and the Flow-guided Feature with stable and flow boxes for proposals hand detection network with tubelet forward optical flow refine model (FGSFP+TFFR) is employed for hand detection in non-key frames. The deformable convolution and sequence attention three-dimensional residual network encoder with gate recurrent unit decoder model (DCSR3D+GRU) is employed for comprehensive correctness discrimination of continuous sign language actions.**

(2 + 1)D convolution for isolated sign language recognition, and Venugopalan and Reghunadhan [1] employed GoogleNet [5] and Bidirectional Long Short-term Memory (BiLSTM) for isolated sign language recognition. In continuous sign language recognition, feature extraction is typically conducted for video frames, and then the hand features are processed by time-series methods. Continuous sign language recognition can recognize several complex sign language actions in the corresponding video data. For example, Min et al. [6] employed 2D and 1D convolution feature extraction and utilized BiLSTM and Connectionist Temporal Classification Loss (CTC Loss) and proposed Visual Alignment Constraint (VAC) for continuous sign language recognition. Furthermore, Pu et al. [7] employed 3D convolution and CTC Loss for continuous sign language recognition. In addition, Papastratis et al. [8] used a Convolutional Neural Network (CNN) to extract features, and they used text embeddings and a Sequence To Sequence (seq2seq) structure for continuous sign language recognition. Xiao et al. [9] employed the Faster Regions with CNN Features (Faster RCNN) [10] to detect the hand in video frames. Here, they fused CNN-extracted features of the detected hand patches and body information, and employed the Spatiotemporal Long Short-term Memory (ST-LSTM) seq2seq structure for continuous sign language recognition.

This study focuses on the category and standardization comprehensive correctness discrimination of sign language actions made by students. As shown in Fig. 1, a hand detection, and continuous sign language correctness discrimination model is proposed. The general sign language recognition method attempts to identify the sign language category; however, there is no standardization discrimination for actions of the same sign language category. The primary contributions of this study are summarized as follows.

A SLCD dataset is constructed. All the demonstrators are making Chinese sign languages. Here, each sign language video has two types of labels, i.e., sign language category, and standardization category simultaneously. Semi-supervised learning is used to make pseudo hand position labels of the video frames.

The proposed FGSFP+TFFR implements a video hand detection model with tubelet that can detect up to two hands in each frame. The model accelerates the network using flow-guided features. Here, proposals are obtained by stable and flow key frame detections, which effectively reduces the number of proposals. In addition, the model can refine the forward optical flow to obtain center point offset refine map for improved accuracy. The model reduces computational costs while getting sufficiently high hand detection result.

The proposed DCSR3D encoder combines 3D convolution and 2D deformable convolution results with residual structures and implements a sequence attention mechanism. The design of DCSR3D model effectively enhances the extraction of spatiotemporal features. The input to the GRU decoder is the concatenation of the corresponding text embedding and encoder features. This allows the model to discriminate the relationship between sign language video made by students and the corresponding learning text obtained from the sign language learning software. The final output is the category and standardization comprehensive correctness discrimination result. The model was tested on the SLCD dataset with hand patches and full images, leading to obtaining good discrimination results compared to the other models.

II. RELATED WORK

A. 3D CONVOLUTION

Video data contain a large number of frames, and there is a strong relation between adjacent frames. In terms of video recognition, many methods have been proposed to accelerate the network or improve recognition accuracy. Differing from a general CNN, the 3D Convolutional Neural Networks (3DCNN) [11] was proposed for feature synthesis of video frame sequences. In addition, improved 3DCNN structures have been proposed. For example, 3D Residual Convolutional Neural Network (R3D) [12] applied the Residual Network (ResNet) [13] structure to 3D convolution. 2D Spatial Convolution Followed By 1D Temporal Convolution (R(2 + 1)D) [14] utilized 2D and 1D convolution to achieve the effect of 3D convolution, which could realize better results with the same number of parameters. Deformable 3D Convolution (D3D) [15] implemented a learnable bias to the sampling position of the 3D convolution to achieve 3D deformable convolution, and Quantized Tensor Train Neural Networks (QTTNet) [16] employed the Tensor Train (TT) format [17] to improve 3D convolution, which greatly reduced the time costs and number of parameters required for 3D convolution networks. In addition, Cao [18] employed 3D depth-wise separable convolution to recognize video gestures, and Zhu et al. [19] employed a pyramid 3D convolution network architecture for gesture recognition.

B. TUBELET FOR VIDEO OBJECT DETECTION

The tubelet method uses the detection results of adjacent frames to improve object detection in the current frame. For example, Kang et al. [20] utilized tubelet proposal networks to detect video objects. Their method created proposals with tubelets across adjacent video frames. However, their method approach became excellent only when it caught objects with small movement in the frame scope. In addition, Tang et al. [21] utilized cubic proposals to detect small tubelets, which were then connected according to the box Intersection Over Union (IOU) in the same frame, and they used the tubelets to update the classification score. The method proposed by Feichtenhofer et al. [22] predicted the offset based on the detection results of the previous frame, and the adjusted boxes were compared with the detection results of the current frame. The Tubelets with Convolutional Neural Networks (T-CNN) method [23] utilized an optical flow to flow the detection results of the adjacent frame and combined the flow boxes with the proposals for the current frame. This method flowed boxes with the average value of the optical flow inside the detection area, which was rather inferior and with rough prediction. The method proposed by Zhang et al. [24] employed an optical flow to flow the detection results of the previous frame to the current frame. Here, the previous frame detection results were used to crop the optical flow image, and the cropped flow images were used as the input to the neural network to obtain the offset of each target. These tubelet methods have been shown to increase video detection accuracy; however, they also involved a large number of proposals for hand detection, which increased computational costs. In contrast, the proposed hand detection model with tubelet in this paper only obtains proposals by stable and flow key frame detections. The model utilizes the optical flow and heat maps of key frame detections to generate center point offset refine maps, generating more accurate flow proposals.

C. SPARSE PROPOSALS

Many object detection methods are based on candidate boxes, which often cover targets by setting a large number of candidate boxes. Such as Region Proposal Network (RPN) was designed in Faster RCNN to select and regress anchor boxes, and Single Shot MultiBox Detector (SSD) [25] preseted a large number of multi-scale candidate boxes. A large number of candidate boxes set in the former or other normal methods are unnecessary. Therefore, designing sparse and effective candidate boxes will greatly reduce network computation and running time. Iterative Grid Based Object Detector (G-CNN) [26] used sparse preset candidate boxes for detection, which required extra detection iteration to make the prediction boxes close to the target position. DEtection TRansformer (DETR) [27] was an object detection method using Transformer [28], which proposed a fixed number of 100 candidate boxes in each image and output the detection results through the decoder. This was actually setting sparse candidate boxes in the image. DeNoising DETR (DN-DETR) [29] designed a denoising training method to enhance the stability of candidate boxes matching with real targets during DETR training. Sparse RCNN [30] directly set sparse proposals in the detection model and designed a cascade decoder for detection. In Dual Attention Sparse R-CNN [31], the dual attention module was applied to each cascade stage to improve detection accuracy. Different from the constant candidate boxes in previous work, Dynamic Sparse R-CNN [32] used Dynamic Proposal Generation (DPG) to generate dynamic sparse candidate boxes. It can be seen that when reducing the number of preset candidate boxes, enhance the target coverage of the candidate boxes or iterative box refine can perform better in the detection results. This paper reduces the number of proposals to a very small extent for hand detection in the sign language videos, and considers both fast and slow hand movements to ensure the hand target coverage.

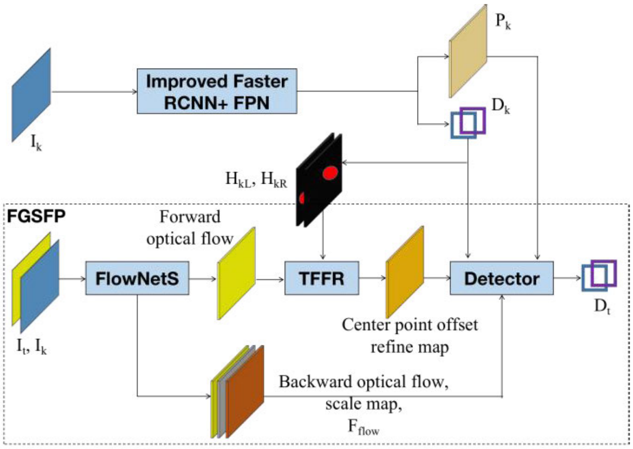
P$_k$
Improved Faster RCNN+ FPN
D$_k$
I$_k$
H$_{kL}$, H$_{kR}$
FGSFP
Forward optical flow
FlowNetS
TFFR
Detector
D$_t$
I$_t$, I$_k$
Center point offset refine map
Backward optical flow, scale map, F$_{flow}$

FIGURE 2. Structure of the proposed video hand detection model. The improved Faster RCNN+FPN is used in key frames, and FGSFP+TFFR is used in non-key frames.

III. PROPOSED METHODS

The proposed hand detection and continuous sign language correctness discrimination model is shown in Fig. 1. The proposed method is to detect the hands first and then send the cropped hand patches to the following correctness discrimination model. Video hand detection model is discussed in Section. A, DCSR3D model encoder is discussed in Section. B.

A. VIDEO HAND DETECTION MODEL

The proposed FGSFP+TFFR model structure used in non key frames and improved Faster RCNN+FPN used in key frame are shown in Fig. 2. The improved Faster RCNN+FPN is discussed in Section I), FGSFP+TFFR is discussed in section II). In Section II), FGSFP+TFFR is separately discussed by a. Optical flow prediction, b. TFFR module, c. Proposals of stable and flow key frame detections, d. Hand detection and post-processing.

1) IMPROVED FASTER RCNN+FPN NETWORK IN KEY FRAMES

The proposed improved Faster RCNN + Feature Pyramid Networks (FPN) [33] model enlarges the FPN C_3, C_4, and C_5 layer features to the size of the C_2 layer for addition. It is similar to the Libra R-CNN [34] but does not separate the features after addition. Here, the feature map P is computed as shown in Equation (1).Here, the interpolation algorithm uses the nearest interpolation algorithm, W_u1, W_u2, and W_u3 represent the resampling operation, and W is a 1×1 convolution kernel weight matrix.

$$P = W(C_2 + W_{u1}(C_3) + W_{u2}(C_4) + W_{u3}(C_5)) \qquad (1)$$

As shown in Fig. 2, the proposed improved Faster RCNN+FPN method is used for the initialization detection of left and right hands in the key frames. Here, the first frame, and frames at every step of n_k where both hands exist are taken as the key frames. Note that frames with only a single hand or without hands are not considered a key frame. In such cases, take the next frame as a candidate key frame until two hands are detected in the frame.

2) FGSFP+TFFR VIDEO HAND DETECTION NETWORK IN NON-KEY FRAMES

As shown in Fig. 2, the proposed FGSFP+TFFR is designed to detect hands in non-key frames. This method utilizes flow-guided features to obtain non-key frame prediction features. The stable and flow boxes of the key frame hand detections are used to obtain the proposals. The main steps are as follows:

a: OPTICAL FLOW PREDICTION

In the proposed method, Optical Flow with Convolutional Networks Simple (FlowNetS) [35] is employed for optical flow prediction, and the input is the current non-key frame image and key frame image. The backward and forward optical flows are obtained simultaneously. The two branches of the predicted optical flow are as follows.

1. The backward optical flow and scale map are generated. These are used to sample and scale the feature P_k of the key frame and obtain the flow-guided feature. F_{flow} is the feature maps to predict optical flow propagated from FlowNetS, it is concatenated with the flow-guided feature to increase the detection result. The prediction feature of the current frame P_t is computed as shown in Equation (2), where W_c is a 3×3 convolution kernel weight matrix, W is the mapping operation based on the backward optical flow, F_B represents the generation of the backward optical flow, S is the scale generation, I_k is the key frame image, and I_t is the current non-key frame image.

$$P_t = [F_{flow}, W_c * W(P_k, F_B(I_k, I_t), S(I_k, I_t))] \qquad (2)$$

2. The forward optical flow is generated. The obtained forward optical flow is first sent to the TFFR module for refine and used to shift the hand center point position of the key frame detection D_k. The details of TFFR module are shown in section b and that of flowing key frame detection are shown in section c. The flow boxes are used as part of the proposals in the current non-key frame. Note that the size of the flow boxes is not changed.

b: TFFR MODULE

The boxes can be directly transformed by the forward optical flow or using the average optical flow value as the previously proposed method [23]. However, direct utilization of the forward optical flow does not yield sufficient accuracy. Due to the inconsistency between the forward optical flow and the key frame hand detection center point positions, the original forward optical flow cannot perfectly shift the box center points. Here, the hand positions detected in the key frame may have a certain deviation, and the FlowNetS method only estimates the optical flow based on the actual image of the two frames. In addition, the characteristics of the optical flow prediction will lead to worse results, e.g., the optical flow will have obvious edges, and incorrect hand center positions may have large differences. In addition, prediction of the optical flow itself may not be completely accurate. Thus, it is necessary to refine the forward optical flow map to obtain the center point offset refine map.

The TFFR module is proposed to obtain the center point offset refine map, which can be used to shift the key frame hand detections more accurately. Fig. 3 shows the network structure of the proposed TFFR module. The TFFR module adopts a layer-by-layer refine structure, which exploits the advantage of the optical flow refine method in Optical Flow with Convolutional Networks (FlowNet) [35] structure. The input to the proposed TFFR module is the concatenation of forward optical flow and the hand detection D_k center point

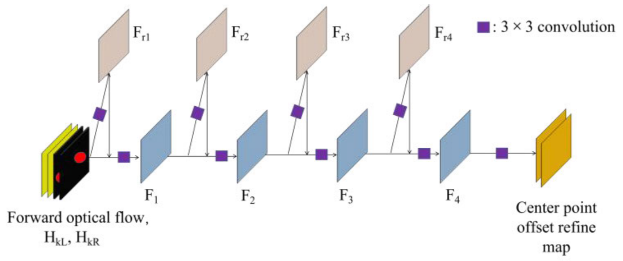

FIGURE 3. Network structure of TFFR module.

heat maps H_{kL} and H_{kR}. Here, the number of input channels is four, and the module comprises four TFFR blocks. Finally, the center point offset refine map is obtained after the final convolution layer.

The TFFR block is calculated as shown in Equation (3), where F_i is the input of the ith block, and F_i+1 is the output feature. F_i+1 is also taken as the input to the next block. F_{ri} is the representation feature. W_s and W_r are 3 × 3 convolution weights, and bn_relu represents the normalization and Rectified Linear Unit (ReLu) activation layers. In each block, the input feature of n_i channels first passes through W_r to obtain the representation feature F_{ri} of k channels. Then, the concatenation feature of F_{ri} and F_i, are going through bn_relu, W_s, and bn_relu layers to obtain the output of n_{i+k} channels. Note that parameter k represents the increased feature number of each TFFR block ($k = 4$ in this paper).

$$F_{i+1} = \text{bn\_relu}(W_{si}(\text{bn\_relu}([F_{ri}, F_i]))) \tag{3}$$

Here, F_{ri} is expressed as follows.

$$F_{ri} = W_{ri}(F_i) \tag{4}$$

The heat maps H_{kL} and H_{kR} are obtained by a Gaussian function centered on the center point of the key frame detection. The heat map generation method used in the proposed method follows that utilized in CenterNet [36]. Equation (5) is used to obtain the pixel value near the center point of the detection box in the heat map after Gaussian processing. In Equation (5), $\tilde{p}$ is the coordinate of the detections, R is the down-sampling ratio, x and y are the pixel positions relative to the center point, and σ_p is the adaptive standard deviation obtained according to the detection size.

$$Y_{xyc} = \exp(-\frac{(x - \widetilde{p}_x)^2 + (y - \widetilde{p}_y)^2}{2\sigma_p^2}), Y \in [0, 1] \tag{5}$$

Here, $\tilde{p}$ is calculated as follows.

$$\widetilde{p} = [\frac{p}{R}] \tag{6}$$

c: PROPOSALS OF STABLE AND FLOW KEY FRAME DETECTIONS

The flowing key frame center point coordinates are calculated as shown in Equation (7), where x_c, and y_c are the x and y coordinates of the flow proposal center point, respectively, F_F is the forward optical flow, I_k is the key frame image, and I_t is the current non-key frame image. Here, x_{ck}, and y_{ck} represent the x and y coordinates of the key frame detection center point, respectively. R is the down-sampling ratio of the forward optical flow graph. In the proposed model, $R = 4$. The measure between the optical flow prediction and the detections is not the same; thus, the value of the center point offset refine map must be multiplied R times. In addition, when sampling the offsets with the center point coordinate value of D_k, the center point coordinate must be reduced R times. Here, x_{dim} and y_{dim} represent the channels corresponding to the x and y offsets in the center point offset refine map, respectively.

$$\begin{aligned} x_c &= x_{ck} + R \cdot \text{TFFR}(F_F(I_k, I_t))_{(x_{ck}/R, y_{ck}/R, x_{dim})} \\ y_c &= y_{ck} + R \cdot \text{TFFR}(F_F(I_k, I_t))_{(x_{ck}/R, y_{ck}/R, y_{dim})} \end{aligned} \tag{7}$$

As shown in Equation (8), in addition to the flow proposals, the stable key frame detections are selected as the proposals in the non-key frame. In summary, there is a total of four proposals in the non-key frames. Note that this considers both fast and slow moving targets comprehensively.

$$B_t = [D_k, B_F] \tag{8}$$

Here, B_F is calculated as follows.

$$B_F = W_B(D_k, F_F(I_k, I_t)) \tag{9}$$

The proposals of non-key frame are obtained by the stable and flow key frame detections; thus, the non-key frame detection network does not utilize RPN and reduce computational costs. The method will also reduce the computation cost of the neural network for the Region Of Interest (ROI) features and the post-processing operation for the final detection results.

d: HAND DETECTION AND POST-PROCESSING

The ROI features are extracted according to the proposals. The features go through two fully-connected layers to obtain the category and the regression value. The detection results of the network are subjected to the following post-processing operations to obtain final left and right hand detection results. First, we retain the left and right hand largest category confidence predictions because only one student is making sign language in each video, and there is at most one effective detection area for the left and right hands. Then, the detections whose category prediction confidence value is less than 0.05 are removed to exclude background area.

B. DCSR3D

The DCSR3D model is designed with 3D convolution and 2D deformable convolution. It is based on the R3D residual structure and added sequence attention mechanism. In this Section, the 2D deformable convolution is discussed in Section I), DCSR3D residual block structure is discussed in Section II).

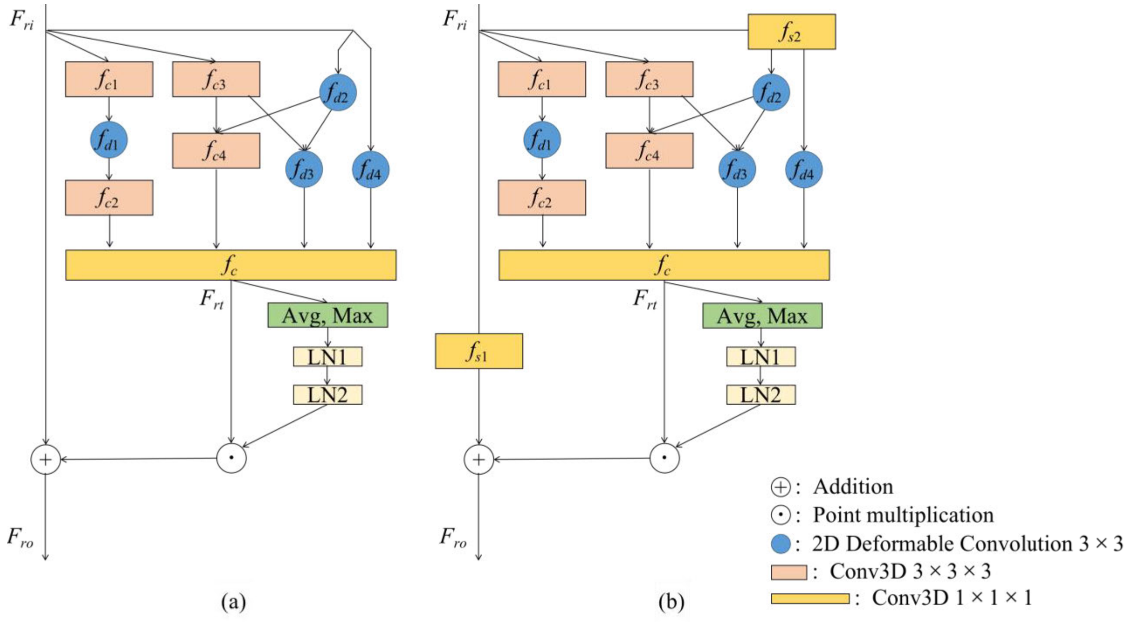

FIGURE 4. Structure of DCSR3D residual block structure, which combines 3D convolution, and 2D deformable convolution. The final result is obtained by the four-path convolution result, and a sequence attention mechanism is implemented. (a) general convolution residual block of DCSR3D; (b) convolution residual block of DCSR3D in down-sampling.

1) DEFORMABLE CONVOLUTION

The deformable convolution method proposed by Dai et al. [37] uses a neural network to predict the sampling position of the convolution kernel to achieve convolution of different shapes. Here, the sampling position differs for each point in the feature map such that the network can select more useful features in the current feature map. Equation (10) shows the computation of a feature point $y(p_0)$.

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{10}$$

Here, p_0 is the feature point position, p_n is the offset of the sampling point in the regular convolution kernel, and Δp_n is the predicted offset of the deformable convolution at each sampling point. In addition, $w(p_n)$ represents the weight of the corresponding position of the convolution kernel and x represents the feature map. In Equation (11), the value sampled in x is calculated according to the bilinear interpolation algorithm.

$$x(p) = \sum_{q} \mathrm{G}(q, p) \cdot x(q) \tag{11}$$

Here, p is the location of the sampling point, q is the spatial location listed around p, and G is the kernel of the bilinear interpolation algorithm. As shown in (12), G comprises two dimensions:

$$\mathrm{G}(q, p) = \mathrm{g}(q_x, p_x) \cdot \mathrm{g}(q_y, p_y) \tag{12}$$

where

$$\mathrm{g}(a, b) = \max(0, 1 - |a - b|) \tag{13}$$

2) DCSR3D RESIDUAL BLOCK

The structure of DCSR3D residual block is shown in Fig. 4. The DCSR3D combines 3D convolution and 2D deformable convolution, which enhances the feature extraction of the current sequence position. The DCSR3D model is based on the basic block of two-layer convolution residual structure from R3D and ResNet. The output feature is obtained from the four-path convolution feature maps. As shown in Equation (14), F_{ri} is the input feature of the residual block and f_c is the $1 \times 1 \times 1$ 3D convolution to synthesis four-path features and to obtain the interval feature F_{rt}. The first path of the four-path convolution adds 2D deformable convolution layer f_d1 between the two 3D convolution layers f_c1 and f_c2, and the second path is used to concatenate the features of the 3D and 2D deformable convolutions f_c3 and f_d2 before the second 3D convolution f_c4. The third path concatenates the features of the 3D and 2D deformable convolutions f_c3 and f_d2 before the second 2D deformable convolution f_d3. Finally, the fourth path takes the 2D deformable convolution f_d4 so that the input can directly pass through it. The number channels of each path is reduced to one-half of the original number (with the exception of the first path, which remains unchanged).

$$\begin{aligned} F_{rt} = f_c([&f_{c2}(f_{d1}(f_{c1}(F_{ri}))), \\ &\times f_{c4}([f_{c3}(F_{ri}), f_{d2}(F_{ri})]), \\ &\times f_{d3}([f_{c3}(F_{ri}), f_{d2}(F_{ri})]), \\ &\times f_{d4}(F_{ri}),]) \end{aligned} \tag{14}$$

The interval feature F_{rt} of DCSR3D block is input to the sequence attention layer. The sequence attention mechanism in the proposed method is based on the attention methods employed in the Squeeze-and-excitation Networks (SE Net) [38] and Convolutional Block Attention Module (CBAM) [39] methods. As shown in Equation (15), the sequence attention mechanism first uses Average (Avg) and Maximum (Max) operations on the last two dimensions of the feature map F_{rt}. The dimensions of the attention interval feature map M_t change to $(N, S, C, 2)$. Then it will flatten feature M_t from the second dimension, causing dimension change to $(N, (S \times C \times 2))$. As shown in Equation (16), M_t is then sent through two fully-connected layer: LN_1 and LN_2. Finally, the sigmoid function σ is used to obtain the attention value M_{attn}. Note that the channel and the sequence dimension are flattened together; thus, rather than being calculated for each sequence, the attention mechanism ultimately reflects the more fine-grained sequence attention.

$$M_{\mathrm{t}} = [\mathrm{AvgPool2d}(F_{rt}),\mathrm{MaxPool2d}(F_{rt})] \tag{15}$$

$$\mathrm{M}_{attn} = \sigma(\mathrm{LN}_2(\mathrm{LN}_1(\mathrm{M}_t))) \tag{16}$$

As shown in Equation (17), the attention value M_{attn} will affect the interval feature F_{rt} as the residual value applied to feature F_{ri}. As shown in Fig.4, (b) two 1 × 1 × 1 3D convolution layers f_s1 and f_s2 are added to adapt the feature map with the output size. The whole input and output sizes of DCSR3D residual block are the same as the corresponding R3D and ResNet structure. DCSR3D-Bx only uses block number instead of layer number to name.

$$F_{ro} = F_{r\mathrm{i}} + F_{rt} \cdot M_{\mathrm{attn}} \tag{17}$$

In the sign language recognition task, there are typically many unnecessary motions. For correctness discrimination task, gesture errors in a few frames may affect the correctness discrimination of the overall sign language action. Also, same mistakes in the frames are convert and occupy a small area. Thus, the attention mechanism is conducive to selecting useful spatiotemporal features in the discrimination task of this study.

IV. EXPERIMENT

The experiment was implemented using Pytorch, and two GeForce RTX 2080Ti GPU were used for training. In addition, Stochastic Gradient Descent (SGD) optimization was adopted for model training. The proposed continuous sign language correctness discrimination model was trained and tested on the SLCD dataset. More information about the SLCD dataset and hand detection and sign language correctness discrimination results are presented in this Section.

A. PARAMETERS

The initial learning rate of the DCSR3D+GRU sign language correctness discrimination model on the SLCD dataset was set to 0.005. The batch size was set to four. The model was trained over 21 epochs, and the learning rate was multiplied by 0.33 every four epochs.

The initial learning rate of the FGSFP+TFFR hand detection network on the SLCD dataset was set to 0.001, the batch size was set to four, and the network was trained over 30 epochs. Here, the learning rate was multiplied by 0.1 every ten epochs.

The evaluation metric used to assess the FGSFP+TFFR hand detection network was the Mean Average Precision (mAP), which represents the detection accuracy of all categories. The result was calculated with coco mAP evaluation method of IOU value 0.5 (mAP50) and IOU value between 0.5 and 0.95 with step of 0.05 (mAP50-95). The evaluation metric used to assess the DCSR3D+GRU network for sign language discrimination is Accuracy (Acc.), where it is directly called discrimination accuracy in the following content.

B. IMPLEMENTATION DETAILS

1) HAND DETECTION NETWORK

ResNet50 [13] was used as the backbone network for feature extraction in the key frames. The input size of the images was 3 × 224 × 320. In addition, in the proposed improved Faster RCNN+FPN, the number channels of FPN output feature maps at each layer was 256. The number of channels of feature map P for ROI pooling was 512 for both key and non-key frames. The size of each extracted ROI was 5 × 5. The ROI features went through two fully-connected layers. The layer dimension number of the two fully-connected layers was 512. Note that the fully-connected layers were not identical for the key frames and non-key frames. When training the FGSFP+TFFR model, two frames were randomly selected with step 10 continuous frames for each video. Proposals of IOU greater than 0.01 with ground truth boxes were regarded as positive samples. All videos in the sign language correctness discrimination training dataset were trained for each epoch. The key frames were sampled every 10 frames for testing. FlowNetS used the pretraining weights trained on the Flying Chairs dataset [35]. FGSFP+TFFR hand detection network in non-key frame was fine-tuned with key frame pretrained improved Faster RCNN model. The pretrained weights of the key frame detection network were using the model weights of semi-supervised learning generation 0 when generating the pseudo hand position labels. As shown in formula (18), FGSFP+TFFR used Smooth L1 Loss $L_{DS}1$for the specific point in the center point offset refine map, used traditional Faster RCNN Cross Entropy Loss L_{DC} and Smooth L1 Loss $L_{DS}2$ in the final classification and regression prediction process.

$$L_D = \lambda_1 L_{DS1} + \lambda_2 L_{DC} + \lambda_3 L_{DS2} \tag{18}$$

2) DCSR3D+GRU

Two types of inputs were used in this experiment: hand patch and full image inputs. The input size of the full image was 3 × 112 × 160 (channel, height and width, respectively) and that of the hand patches was 6 × 32 × 32 (channel, height, and width, respectively), i.e., the concatenation of the left and right hand patches. When training and testing the proposed model, 30 frames were selected from the

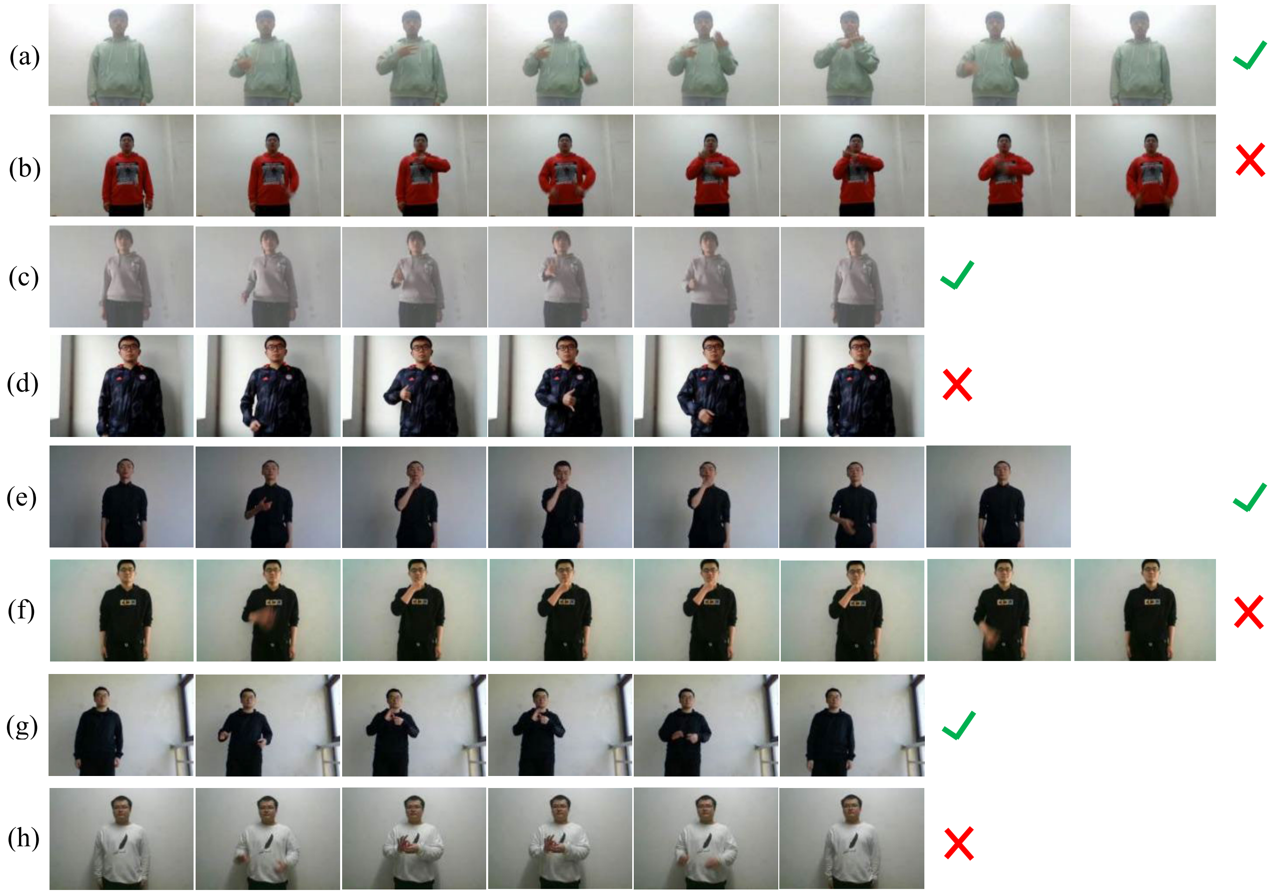

FIGURE 5. Examples of standardization labels of the sign language videos in the SLCD dataset (representative frames are displayed).

beginning to the end of each video with the same step. The DCSR3D-B8 model was trained on the SLCD dataset. Note that the DCSR3D-B8 model had the same number of block, input channels, and output channels as the ResNet18 model. The DCSR3D-B8 encoder output 1,280 dimensions. The sign languages discriminated in this study included a total of 95 Chinese characters, and one additional character was added to represent Chinese characters not realized in this study. The 96 characters were embedded to 32 dimensions and were sent through two fully-connected layers to output 500 dimensions. Here, dropout of 0.3 was added to the text embedding results to prevent overfitting. As shown in Fig. 1., the image encoding features of DCSR3D and text embedding features were concatenated as the input of GRU decoder. The input dimension of the GRU decoder was set to 1,780, and the output and hidden state dimensions were set to 1,280. The GRU decoder output was sent through the fully-connected layer to obtain two output units, i.e., the correct and incorrect classifications. The Cross Entropy Loss L_{Chand} was used to train the DCSR3D+GRU model.

C. SLCD DATASET

An SLCD dataset was collected to facilitate sign language education. Each video included two types of recognition labels, i.e., the sign language category label and the standardization category label. The standardization category label discriminated standardization for actions of the same sign language category. In contrast, general sign language datasets only had standard sign language actions and they could only make models to classify similar sign language actions to the same category ignoring minor defects. Each image also included pseudo hand position labels generated by the improved Faster RCNN+FPN semi-supervised learning method. The videos in the SLCD dataset were obtained using individual cameras. In the data collection process, 76 students were recruited to record sign language videos. Each student performed the same sign language multiple times to ensure sufficient diversity in the dataset. There were 52 Chinese isolate sign languages, and 27 Chinese continuous sign languages. We acquired a total of 20,792 sign language videos. Every second frame of each video was saved as an image, and a total of 1,054,598 images were acquired. The training and testing sets were formed at a ratio of 9:1.

1) CORRECTNESS DISCRIMINATION LABELS

In addition to sign language category labels usually contained in the general sign language datasets, each sign language video was signed with a standardization label. If the sign language video standardization label was standardized, the action was the corresponding sign language action, and the sign language action in the video was considered standard. In this evaluation, incorrect finger positions, incorrect hand movements, or lack of necessary body movements were considered nonstandard actions. As shown in Fig. 5, (a) and (b), (c) and (d), (e) and (f), and (g) and (h) show sign language gestures for "job," "go," "like," and "contact," respectively, where (a), (c), (e), and (g) show standard examples, whereas (b), (d), (f), and (h) show nonstandard examples. The nonstandard parts are described as follows: (b): the thumbs are held up in frames 3 and 6; (d): in frames 3 and 4, the hand should reach out vertically (not horizontally); (f): in frames 3, 4, 5, and 6, the head should nod slightly rather than pinching two fingers twice; (h): in frames 3 and 4, the middle, ring, and little finger of the left hand should be clenched rather than half open.

Note that the original data standardization label only contained correctness correspondence between the sign language videos and corresponding category texts. In this study, the data were doubled to add correctness correspondence between the sign language videos and other category texts. These additional videos were all signed incorrect for the standardization label. The total number of videos used for training and testing for sign language correctness discrimination was 41,584

2) HAND POSITION LABELS WITH IMPROVED FASTER RCNN+FPN SEMI-SUPERVISED LEARNING

The improved Faster RCNN+FPN semi-supervised learning method was used to generate pseudo hand position labels for the video frames. Here, 2,077 representative images were annotated manually with ground truth position labels using the LabelImg software. Among the annotated images, 158 images were selected from the Chinese Sign Language Recognition Dataset [40] to prevent data overfitting.

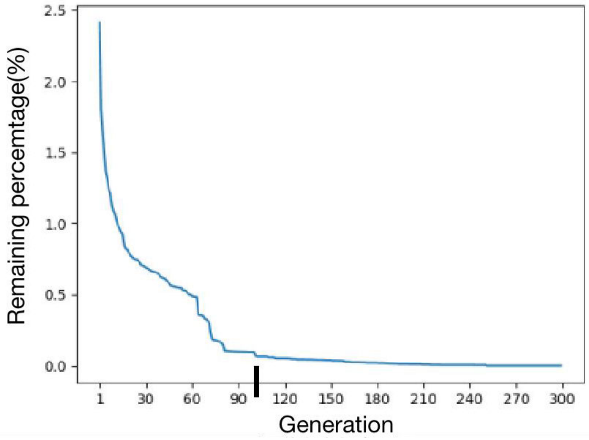

FIGURE 6. **Percentage of remaining images in the improved Faster RCNN+FPN semi-supervised learning process corresponding to each generation, where 289 images were manually labeled in the 100th generation.**

In this evaluation, the improved Faster RCNN+FPN was first trained using the ground truth position labels for 100 epochs. Here, the initial learning rate was set to 0.01, which was multiplied by 0.33 at the thirtieth and sixtieth epochs. Note that the mAP value of these trained images with an IOU value of 0.5 was 98.9%. Second, the improved Faster RCNN+FPN was used to detect the hand area of the stored images. The detection results of both hands with confidence greater than 98% or less than 2% (i.e., no corresponding target) were added to the pseudo position labels. Third, a total of 300 generations was performed to generate pseudo position labels. Each generation training process contained three epoch training, and then the latest trained model weights were used to generate the pseudo position labels of the unlabeled images. Each generation was trained with 10,000 randomly selected images with pseudo hand position labels and all the images with the annotated ground truth position labels. The percentage of remaining images in each generation is shown in Fig. 6. As the remaining unlabeled image percentage declined hardly from approximately the 80th generation, 289 images were manually labeled and added to the training process of each generation from the 100th generation to accelerate the labeling process.

D. HAND DETECTION RESULTS

1) COMPARISON WITH DIFFERENT DETECTION MODELS

The FGSFP+TFFR model was compared with some other detection models on SLCD dataset, i.e., traditional object detection models: Faster RCNN+FPN, SSD; video object detection methods: RDN [41], MEGA [42]; video object detection models with flow guided feature: DFF [43], FGFA [44]; detection models with sparse proposals: Sparse R-CNN, DETR, Anchor DETR [45], DN-DETR. Most of the models used ResNet50 backbone in the detection or the key frame detection. FGSFP+TFFR key frame detection model was using pretrained weights of labeled model generation 0. The labeled model generation 0 is the improved Faster RCNN+FPN model after 100 epochs training with the ground truth position labels.

Table 1 is the mAP detection results comparison of different models. It can be seen traditional object detection methods and the video object detection methods had similar mAP detection results. Where RDN achieved highest mAP50 detection results of 99.2% compared to other detection methods. The results with DETR method structure had achieved higher mAP50-95 detection results. Sparse R-CNN with cascade decoder achieved highest mAP 50-95 detection result of 84.6% compared to other detection methods. The detection result of FGSFP+TFFR designed in this paper was 78.9% for mAP50-95 and 99.0% for mAP50. FGSFP+TFFR model had achieved the ideal result compared to other methods. The result shows that acquiring proposals via stable and flow key frame detections is effective. The result also shows the forward optical flow to flow boxes is adaptive to flow guided feature structure. FGSFP achieved detection result of 77.1%

TABLE 1. Comparison of mAP detection results of different methods.

Detection network	Backbone/ key frame backbone	mAP50-95 (%)	mAP50 (%)	FPS
Faster RCNN+FPN	ResNet50	68.5	97.9	55.9
SSD	VGG16	74.2	98.8	76.9
DFF	ResNet50	65.0	98.6	59.6
FGFA	ResNet50	66.4	98.8	21.9
RDN	ResNet50	72.2	**99.2**	9.1
MEGA	ResNet50	70.5	97.5	7.2
Sparse R-CNN	ResNet50	**84.6**	99.0	-
DETR	ResNet50	78.1	97.4	31.9
Anchor DETR	ResNet50	82.0	99.0	22.9
DN-DETR	ResNet50	83.8	99.0	-
Improved Faster RCNN+FPN(ours)	ResNet50	70.4	97.9	54.4
FGSFP+TFFR (FlowNet1/4)	ResNet50	71.7	95.8	**77.5**
FGSFP+TFFR (FlowNet1/2)	ResNet50	74.6	97.4	73.5
FGSFP	ResNet50	77.1	98.0	62.1
FGSFP+TFFR	ResNet50	78.9	99.0	61.4

for mAP50-95 and 98.0% for mAP50. It was slightly lower than FGSFP with TFFR module. This shows the effectiveness of TFFR module design.

Detection models were tested with the running time. It can be seen MEGA and RDN got the lowest FPS, for they concentrated more on accuracy but not speed. SSD detection methods with one-stage structure got higher speed compared to other methods. DFF and FGFA were using FlowNetS to generate optical flow, it can be seen DFF got little higher speed than Faster RCNN+FPN. DETR got FPS 31.9, Anchor DETR got FPS 22.9. Because Sparse R-CNN and DN-DETR were trained on WSL2 system, their model running speed were not shown in Table 1. From their original research experiment we can know the running speed of DN-DETR is a little higher than DETR and Sparse R-CNN is a little lower than DETR. FGSFP+TFFR got FPS of 61.4, FGSFP got FPS of 62.1. TFFR module caused little FPS decline but got remarkable detection result increased. FGSFP+TFFR was also tested with different FlowNetS scale. FlowNet scale 1/2 and 1/4 remained the original FlowNetS structure but used less convolution kernels in each layer. It can be seen FlowNetS from scale 1 to 1/4 got speed increased and with acceptable detection result decreased. FGSFP+TFFR with FlowNetS of scale 1/4 got the highest FPS of 77.5.

2) ABLATION STUDY

TABLE 2. FGSFP+TFFR ablation experiment results.

TFFR	F_{flow}	Fine-tune	L_{D1}	mAP50-95 (%).
	√	√		60.6
√	√	√		61.2
	√		√	71.3
√	√		√	72.5
		√	√	75.6
√		√	√	78.0
	√	√	√	77.1
√	√	√	√	**78.9**

Table 2 compares the mAP50-95 results of different FGSFP+TFFR model structures and training strategies with the full weights FlowNetS. It can be seen for each paired TFFR result comparisons, add TFFR module would definitely increase the detection result. This is because TFFR adds key frame detection information and solves inconsistency between the forward optical flow and the key frame detection center point positions. For FGSFP+TFFR and FGSFP model results, add F_{flow} would increase the detection results. Although flow guided feature can be used to detect objects in the non key frames, F_{flow} also has abundant feature information and is helpful to detect object in the non key frames. When fine-tuned the non key frame detection models, it would not change the key frame model weights. For the labeled model generation 0 detection result was relatively high and non key frame model highly relays on the key frame detections, results with fine-tune training method would get higher detection results. L_D1 was added to supervise center point offset refine map to generate more useful proposals, for some of the proposal deviate too much would not be select in the ROI process when training. It can be seen L_D1 would definitely increase the detection results for FGSFP+TFFR and FGSFP model.

TABLE 3. Comparison of correctness discrimination results of different encoder models.

Encoder	Backbone	Full image acc. (%)	Hand patches acc. (%)
R3D	ResNet18	73.507	79.624
TSN	ResNet18	70.713	77.553
TSM	ResNet18	70.689	77.697
C3D	-	69.870	69.894
R(2+1)D	ResNet18	70.857	78.348
DCSR3D-B8	-	**73.868**	**81.647**

E. SIGN LANGUAGE CORRECTNESS DISCRIMINATION RESULTS

1) COMPARISON WITH DIFFERENT ENCODER MODELS

Table 3 compares the correctness discrimination results of different encoder models. As can be seen, the discrimination result obtained using the DCSR3D-B8 encoder with hand patches was 81.647%, and the discrimination result obtained using full image input was 73.868%, respectively representing the best results. The proposed method was able to obtain this level of performance because it implements deformable convolution to better synthesize the features of the current sequence. In addition, the proposed method implements the sequence attention mechanism to select more useful sequence features. As can be seen, the discrimination accuracy of C3D was the lowest, which indicates that the direct feature fusion technique cannot obtain good results on the target dataset. The discrimination results of R3D were 73.507% for full image and 79.624% for hand patches accuracies, which were good compared to the other methods; however, as demonstrated by Hara et al. [12] in their experiments, this method did not perform well on the Kinetics dataset [33]. It is found that R3D structure synthesizes features while checks out sign language action failures from the specific sequence position with short cut path more easily on the sign language discrimination dataset. The results obtained by combining general 2D and 1D convolutions in R(2+1)D discrimination results provided 70.857% for full image and 78.348% for hand patches accuracies, which were not as good as the results obtained by the proposed model. In addition, the proposed method outperformed the Temporal Segment Networks (TSN) [34] and Temporal Shift Module (TSM) [35] models, which focus on 2D convolution with discrimination results 70.713% and 70.689 for full image accuracy and 77.553 and 77.697 for hand patches accuracy. In the experimental dataset described in this paper, the discrimination results obtained by different encoders exhibited large differences. Note that the discrimination results obtained using the full image input and hand patch input also exhibited large differences. These results indicate that the complexity of the SLCD dataset is high.

TABLE 4. **DCSR3D-B8 ablation experiment results.**

Sequence attention	2D deformable convolution feature concatenation	Hand patch acc. (%)
		80.588
√		81.503
	√	80.780
√	√	**81.647**

2) ABLATION STUDY

An ablation experiment was conducted, and Table 4 shows the experimental results for the DCSR3D-B8 model. Note that these displayed results were obtained using hand patches. Table 4 also shows results obtained with and without adding the sequence attention mechanism to the final output feature of each block. The 2D deformable convolution feature concatenation indicates whether to add the convolution feature results of paths 2, 3, and 4 in the DCSR3D block. As can be seen, the DCSR3D-B8 model with the sequence attention mechanism and 2D deformable convolution feature concatenation obtained the highest discrimination result. The discrimination results of other structures that reduce the components are not as effective as the proposed DCSR3D-B8 model structure.

V. CONCLUSION

Sign language education software requires correctness discrimination not only for different sign language categories, but also for sign language standardization of the same sign language category. The students can practice specific sign language actions using the sign language discrimination method realized in this study in a sign language education software until students can perform perfectly.

In this study, an SLCD dataset, which includes sign language category and standardization category labels, was collected. In addition, the improved Faster RCNN+FPN semi-supervised learning method was employed to make pseudo left and right hand position labels.

The proposed detection method was evaluated experimentally, the FGSFP+TFFR was fine-tuned with pretrained key frame detection model. The FGSFP+TFFR method got sufficiently high detection result while reducing computational costs using flow-guided features and fewer proposals. The proposed method uses TFFR to obtain accurate flow proposals. The proposed detection method is specifically designed for one or two hand detections in the target video data.

In addition, the DCSR3D+GRU model was designed to realize comprehensive correctness discrimination of the sign language category and standardization. The proposed DCSR3D model performs better in feature synthesis of spatiotemporal, for it implements the fine-grained sequence attention mechanism and with the full use of 2D deformable convolution. The combination of 3D convolution and 2D deformable convolution enriches the interval feature. R3D was found to be better suited for recognizing mistakes compared to the general recognition task.

REFERENCES

[1] A. Venugopalan and R. Reghunadhan, “Applying deep neural networks for the automatic recognition of sign language words: A communication aid to deaf agriculturists,” *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115601, doi: 10.1016/j.eswa.2021.115601.

[2] W. Zhang, “Study on gesture language translation system based on neural network,” M.S. thesis, Dept. ECE, Jiangxi Univ. Finance Econ., Nanchang, China, 2018.

[3] S. Basiri, A. Taheri, A. Meghdari, and M. Alemi, “Design and implementation of a robotic architecture for adaptive teaching: A case study on Iranian sign language,” *J. Intell. Robot. Syst.*, vol. 102, no. 2, p. 48, May 2021, doi: 10.1007/s10846-021-01413-2.

[4] F. Wang, Y. Du, G. Wang, Z. Zeng, and L. Zhao, “(2+1)D-SLR: An efficient network for video sign language recognition,” *Neural Comput. Appl.*, vol. 34, no. 3, pp. 2413–2423, Feb. 2022, doi: 10.1007/s00521-021-06467-9.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[6] Y. Min, A. Hao, X. Chai, and X. Chen, “Visual alignment constraint for continuous sign language recognition,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 11522–11531, doi: 10.1109/ICCV48922.2021.01134.

[7] J. Pu, W. Zhou, and H. Li, “Dilated convolutional network with iterative optimization for continuous sign language recognition,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Wellington, New Zealand, Jul. 2018, pp. 885–891, doi: 10.24963/ijcai.2018/123.

[8] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, and P. Daras, "Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space," *IEEE Access*, vol. 8, pp. 91170–91180, 2020, doi: 10.1109/ACCESS.2020.2993650.

[9] Q. Xiao, X. Chang, X. Zhang, and X. Liu, "Multi-information spatial–temporal LSTM fusion continuous sign language neural machine translation," *IEEE Access*, vol. 8, pp. 216718–216728, 2020, doi: 10.1109/ACCESS.2020.3039539.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.

[12] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6546–6555, doi: 10.1109/CVPR.2018.00685.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[14] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6450–6459, doi: 10.1109/CVPR.2018.00675.

[15] X. Ying, L. Wang, Y. Wang, W. Sheng, W. An, and Y. Guo, "Deformable 3D convolution for video super-resolution," *IEEE Signal Process. Lett.*, vol. 27, pp. 1500–1504, 2020, doi: 10.1109/LSP.2020.3013518.

[16] D. Lee, D. Wang, Y. Yang, L. Deng, G. Zhao, and G. Li, "QTTNet: Quantized tensor train neural networks for 3D object and video recognition," *Neural Netw.*, vol. 141, pp. 420–432, Sep. 2021, doi: 10.1016/j.neunet.2021.05.034.

[17] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011, doi: 10.1137/090752286.

[18] F. Cao, "Research on dynamic gesture recognition based on deep learning," M.S. thesis, Dept. Signal Inf. Process., Xi'an Univ. Technol., Xi'an, China, 2021.

[19] G. Zhu, L. Zhang, L. Mei, J. Shao, J. Song, and P. Shen, "Large-scale isolated gesture recognition using pyramidal 3D convolutional networks," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Cancún, Mexico, Dec. 2016, pp. 19–24, doi: 10.1109/ICPR.2016.7899601.

[20] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang, "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 889–897, doi: 10.1109/CVPR.2017.101.

[21] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1272–1278, May 2020, doi: 10.1109/TPAMI.2019.2910529.

[22] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 3057–3065, doi: 10.1109/ICCV.2017.330.

[23] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018, doi: 10.1109/TCSVT.2017.2736553.

[24] J. Zhang, S. Zhou, X. Chang, F. Wan, J. Wang, Y. Wu, and D. Huang, "Multiple object tracking by flowing and fusing," 2020, *arXiv:2001.11180*.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 9905, Amsterdam, The Netherlands, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[26] M. Najibi, M. Rastegari, and L. S. Davis, "G-CNN: An iterative grid based object detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2369–2377, doi: 10.1109/CVPR.2016.260.

[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, in Lecture Notes in Computer Science, vol. 12346, Glasgow, U.K., 2020, pp. 213–229, doi: 10.1007/978-3-030-58452-8_13.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, vol. 30, 2017, pp. 1–11.

[29] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 13609–13617, doi: 10.1109/CVPR52688.2022.01325.

[30] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 14449–14458, doi: 10.1109/CVPR46437.2021.01422.

[31] S. Park, S. Lee, J. Kang, S. Park, S. Choi, and J. Paik, "Dual-attention sparse R-CNN via single ROI transformer and dynamic CBAM," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, Yeosu, South Korea, Oct. 2022, pp. 1–3, doi: 10.1109/ICCE-Asia57006.2022.9954801.

[32] Q. Hong, F. Liu, D. Li, J. Liu, L. Tian, and Y. Shan, "Dynamic sparse R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 4713–4722, doi: 10.1109/CVPR52688.2022.00468.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[34] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 821–830, doi: 10.1109/CVPR.2019.00091.

[35] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2758–2766, doi: 10.1109/ICCV.2015.316.

[36] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[37] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 764–773, doi: 10.1109/ICCV.2017.89.

[38] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[39] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, vol. 11211, Munich, Germany, 2018, pp. 3–9, doi: 10.1007/978-3-030-01234-2_1.

[40] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, "Chinese sign language recognition with adaptive HMM," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Seattle, WA, USA, Jul. 2016, pp. 1–6, doi: 10.1109/ICME.2016.7552950.

[41] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 7022–7031, doi: 10.1109/ICCV.2019.00712.

[42] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 10334–10343, doi: 10.1109/CVPR42600.2020.01035.

[43] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4141–4150, doi: 10.1109/CVPR.2017.441.

[44] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 408–417, doi: 10.1109/ICCV.2017.52.

[45] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, Palo Alto, CA, USA, 2022, pp. 2567–2575, doi: 10.1609/aaai.v36i3.20158.

[46] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. ECCV*, in Lecture Notes in Computer Science, Amsterdam, The Netherlands, 2016, pp. 20–36, doi: 10.1007/978-3-319-46484-8_2.

[48] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 7082–7092, doi: 10.1109/ICCV.2019.00718.

MENGLIN ZHANG received the B.S. degree in electronic science and technology from the Hefei University of Technology, Hefei, China, in 2019, and the M.S. degree in software engineering from the Tianjin University of Technology, Tianjin, China, in 2023. His current research interests include image processing, machine learning, and computer vision.

SHUYING YANG received the B.S. degree in industrial electrical automation from the Tianjin University of Technology and Education, China, in 1982, the M.S. degree in multimedia technology from Tianjin Normal University, China, in 2001, and the Ph.D. degree in computer science from Tianjin University, China, in 2008. Her current research interests include pattern recognition, machine learning, and time series. She is a member of the China Computer Federation.

MIN ZHAO received the B.S. degree in computer science and technology from Taiyuan Normal University, Shanxi, China, in 2019, and the M.S. degree in computer technology from the Tianjin University of Technology, Tianjin, China, in 2022. Her current research interests include pattern recognition and computer vision.