Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 21(3): 3594–3617.
DOI: 10.3934/mbe.2024158
Received: 30 November 2023
Revised: 19 January 2024
Accepted: 30 January 2024
Published: 05 February 2024

Research article

Research on gesture recognition algorithm based on MME-P3D

Hongmei Jin, Ning He*, Boyu Liu and Zhanli Li

College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

* **Correspondence:** Email: 19208088023@stu.xust.edu.cn.

Abstract: A Multiscale-Motion Embedding Pseudo-3D (MME-P3D) gesture recognition algorithm has been proposed to tackle the issues of excessive parameters and high computational complexity encountered by existing gesture recognition algorithms deployed in mobile and embedded devices. The algorithm initially takes into account the characteristics of gesture motion information, integrating the channel attention (CE) mechanism into the pseudo-3D (P3D) module, thereby constructing a P3D-C feature extraction network that can efficiently extract spatio-temporal feature information while reducing the complexity of the algorithmic model. To further enhance the understanding and learning of the global gesture movement's dynamic information, a Multiscale Motion Embedding (MME) mechanism is subsequently designed. The experimental findings reveal that the MME-P3D model achieves recognition accuracies reaching up to 91.12% and 83.06% on the self-constructed conference gesture dataset and the publicly available Chalearn 2013 dataset, respectively. In comparison with the conventional 3D convolutional neural network, the MME-P3D model demonstrates a significant advantage in terms of parameter count and computational requirements, which are reduced by as much as 82% and 83%, respectively. This effectively addresses the limitations of the original algorithms, making them more suitable for deployment on embedded and mobile devices and providing a more effective means for the practical application of hand gesture recognition technology.

Keywords: computer vision; image processing; gesture recognition; P3D convolution; deep learning; attention mechanism

1. Introduction

With the continuous advancement of computer science, human-computer interaction and communication with various intelligent devices have become integral aspects of daily life [1]. Gesture language, as a unique form of communication, has garnered widespread attention due to its natural and intuitive characteristics. Although it may not be as convenient as spoken communication, gestures can

still accurately convey users' emotional information. As an indispensable key technology for future human-computer interaction, vision-based gesture recognition technology has emerged as a current research focus. However, the specificity, diversity, and polysemy of gestures themselves, coupled with the complexity of the human hand structure and limitations in computer vision technology, have made vision-based gesture recognition a challenging research domain that has attracted numerous researchers to dedicate their efforts to it [2].

Gesture recognition techniques can be broadly classified into two primary categories: wearable sensor-based and computer vision-based. Initially, gesture recognition methods relied on electromagnetic gloves and other wired devices directly connected to the computer. In this approach, hand information was transmitted to the computer recognition system for further processing. Xue et al. [3] employed Cyber Glove data gloves in conjunction with a hybrid method to identify ten distinct types of gestures. Although this method demonstrated a high degree of accuracy in gesture detection, its practical applicability is constrained due to factors such as the costly nature of data gloves and the cumbersome wearing process [4]. In 2020, Zhang et al. [5] developed a flexible wearable data glove for acquiring human gesture data and employed a radial basis function neural network for gesture capture and recognition, achieving 88.73% recognition accuracy. In contrast, vision-based gesture recognition technology has gradually reduced its dependence on hardware devices since its development began in the 1990s. Dardas et al. [6] addressed the challenge of gesture tracking and recognition in complex scenarios by extracting hand keypoints using SIFT features and SVM classifiers and training a model to recognize ten different gestures. However, this method requires keypoint extraction prior to recognition, which is less efficient to execute and often necessitates the design of some effective feature extraction schemes to enhance gesture recognition performance.

The progression of computer hardware and software has facilitated the extensive application of deep learning [7], which has also opened up new avenues of exploration in the field of gesture recognition. A prior scholarly investigation by Barros et al. [8] proposed a multi-channel convolutional neural network model for real-time gesture recognition that enhanced the classification features with a cubic convolutional kernel. Gnanapriya et al. [9] proposed an enhanced two-stage integrated model combining U-NET and convolutional neural networks for gesture segmentation and recognition. With the progression of technological tools, research on gesture recognition methods has also made substantial strides. Miao et al. [10] employed the ResC3D convolutional neural network in addressing dynamic gesture recognition, merging the advantages of the residual network and the 3D convolutional neural network [11]. This approach can extract spatio-temporal features while learning deep information. In the ChaLearn LAP of 2017 [12], it achieved favorable results in the multimodal isolated gesture recognition challenge. Wang et al. [13] employed gesture contour features extracted using the slope difference distribution (SDD) method for recognition. Initially, the hand contour was extracted, followed by the calculation of the peaks and valleys of the hand contour through the SDD algorithm for model matching recognition. Gao et al. [14] improved the 2D hand pose estimation based on the OpenPose method, developed a fast 3D hand pose estimation approach, and utilized a weighted fusion method to combine RGB, depth, and 3D skeleton data of the gesture. Finally, they employed the 3DCNN+ConvLSTM framework to recognize and classify the combined dynamic gesture data, effectively enhancing the recognition performance. However, the algorithmic model of this method is large and unsuitable for deployment on mobile and embedded devices. Li et al. [15] utilized millimeter waves for gesture recognition and devised a data enhancement framework to compute the

correlation between signal and gesture changes. They also segmented the gesture image to enhance computational efficiency. The method extracts spatio-temporal information from dynamic windows for gesture recognition, further advancing the development of gesture recognition technology. Currently, behavior recognition techniques rooted in skeleton data have achieved commendable recognition outcomes [16–20], as they provide granular details concerning the positioning of human joints and movement trajectories. This attribute is particularly instrumental for the accurate recognition of intricate and continuous actions. However, it should be noted that the domain of gesture recognition is inherently more circumscribed compared to the broader scope of behavior recognition. Consequently, temporal sequences extracted using information from the skeleton network tend to exhibit a lesser degree of variation. Therefore, directly transplanting these methods onto gesture recognition tasks may encounter certain inherent limitations.

Since convolutional neural networks (CNNs) have demonstrated remarkable accuracy in the domain of image classification tasks, a growing number of researchers have ventured into investigating their application to video understanding, particularly within the realm of gesture recognition. Although both motion recognition and image classification are fundamentally classification problems, they present numerous challenges and intricacies when dealing with sequential video frames due to the distinct natures of video data and the differing types of feature information that must be extracted. In action recognition contexts, it is imperative not only to consider spatial feature details within each video frame—such as hand position, hand morphology, and environmental features—but also temporal dynamics between frames, including the kinematic trend of the hand movement. This necessitates a holistic approach that captures both spatial and temporal aspects effectively.

In this study, we propose the MME-P3D algorithm. Firstly, a P3D-C network is designed for end-to-end gesture recognition. This network combines the channel attention mechanism CE with the P3D network to model the channel relationship of input features, obtaining the channel information weight distribution of the features. By strengthening useful channel features and suppressing irrelevant ones, it enhances the feature extraction capability of the P3D-C network. Subsequently, the Multiscale-Motion Excitation (MME) mechanism for pooling motion attention is integrated into the P3D-C network. Explicit motion features are constructed by computing the feature differences between two adjacent frames, focusing on and extracting the temporal feature information throughout the entire gesture movement process. This enables the algorithmic model to better understand and learn the dynamic information during gesture movement, significantly improving the accuracy and efficiency of gesture recognition. The primary contributions of the MME-P3D-based gesture recognition algorithm are briefly summarized as follows:

1) We designed a feature extraction network, P3D-C, tailored to the characteristics of gesture motion information. The network employs a pseudo-3D convolution structure to simulate 3*3*3 convolution for spatio-temporal feature extraction, effectively reducing the number of parameters. Concurrently, we integrated the channel attention (CE) mechanism into the P3D convolution to further enhance the feature extraction capability of the P3D convolution block.

2) We developed a multi-scale motion attention mechanism, MME, that constructs explicit motion features by computing feature differences between adjacent frames. This significantly reduces the number of parameters and computation required by the model while substantially improving the performance and efficiency of gesture recognition.

The remaining sections of the article are organized as follows: The first part presents a review

of the state-of-the-art research related to gesture recognition. The second part briefly introduces the lightweight technology of convolutional neural network. The third part provides a detailed description of the MME-P3D gesture recognition algorithm. The fourth part showcases and analyzes the results of comparative experiments. Lastly, the fifth part summarizes the algorithm.

2. Related works

Gesture recognition is essentially a form of image classification that necessitates two pivotal stages: feature extraction and subsequent classification. In the initial phase, the extraction process entails discerning critical attributes that distinctly characterize a gesture, such as contours, textures, and colors, from the input visual data. The derived features are then subjected to classification in order to differentiate between various types of gestures. The CNNs have emerged as a pivotal technology for achieving both efficient and accurate gesture recognition in this domain.

With the pervasive use of mobile and embedded devices, deploying CNNs on edge devices holds substantial practical significance and relevance. However, the relentless pursuit of heightened recognition precision and performance has led to an increasing depth in network model layers, escalating complexity, surging numbers of parameters, and computational demands. Consequently, there's a decrease in the inference speed of these models, along with a substantial occupation of memory and computational resources. Given the inherently constrained computational and storage capacities typical of mobile and embedded systems, deploying these resource-intensive models proves challenging, limiting their applicability and impeding widespread adoption. Thus, striking an optimal balance among accuracy, inference speed, and model size becomes imperative. This has rendered the adaptation of convolutional neural network structures a pressing research topic in the academic community. In recent years, numerous research endeavors have focused on reducing the number of parameters and operations in models by optimizing 3D convolutional structures. Xu et al. [21] proposed an online lightweight two-stage framework for accurate detection and classification of dynamic gestures for a single RGB camera on raw video streams in real scenarios, which solves the challenge of fast and accurate recognition of gestures in real systems. Qiu et al. [22] introduced a pseudo-3D convolutional network that employs a pseudo-3D convolutional structure to simulate 3D convolutional operations, effectively addressing the issue of oversized network models caused by traditional 3D convolution. This improvement enhances the efficiency and performance of classification and recognition tasks, with a multitude of experimental results verifying the validity and feasibility of the pseudo-3D convolutional structure. Moreover, deep learning models such as R (2+1) D [23] and S3D [24] have also been extensively analyzed in numerous experiments. These studies demonstrate that it is feasible to decompose the 3D convolution operation into a 2D convolution in the spatial dimension and a 1D convolution in the temporal dimension, thereby combining spatial feature information with temporal feature information. This approach significantly reduces the number of parameters and computational complexity, improving the algorithmic model's efficiency and enhancing the network's robustness. However, while converting the convolution operation can effectively reduce the number of parameters and improve computational efficiency, due to the limitations of kernel size, this optimized neural network can only extract encoded short-term motion feature information within a small and fixed-length time-domain window. As a result, it is difficult to obtain the complete time-series motion feature information for the entire action process, which may decrease gesture recognition accuracy to some extent.

3. MME-P3D gesture recognition algorithm

The MME-P3D gesture recognition algorithm is mainly composed of CE-P3D convolution and ME attention mechanisms, combined with classification tasks for training and optimization. To reduce the number of parameters and computation in the algorithm model, we employ a P3D convolution kernel to simulate 3D convolution for extracting spatio-temporal features of gesture actions. Subsequently, global spatio-temporal information is modeled through multi-scale channel attention to strengthen valid information while suppressing invalid information, thereby enhancing the feature extraction capability of the algorithmic network. Furthermore, we integrate the MME as an adjunct, which constructs explicit motion features by calculating feature differences between adjacent frames. This aids the algorithmic model in better understanding and learning dynamic information during gesture movement. The overall architecture of this network model is depicted in Figure 1.

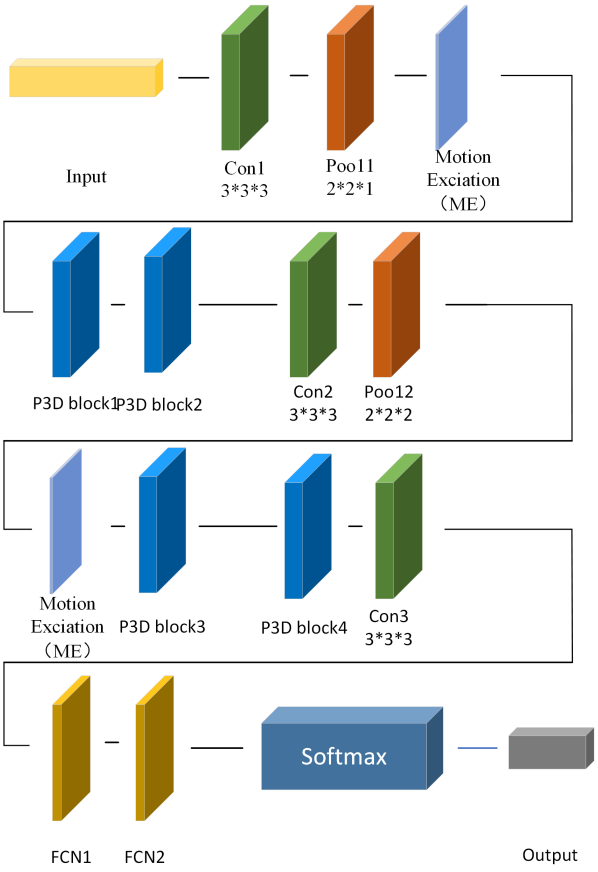


Figure 1. Overall framework diagram of MME-P3D gesture action recognition network.

3.1. P3D-C network

In this paper, a feature extraction network (P3D-C network) is constructed using P3D convolution. The structure of the network is schematically depicted in Figure 2 and comprises four P3D convolution blocks.

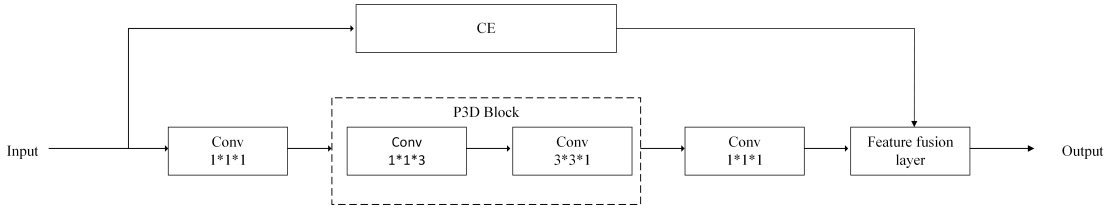


Figure 2. P3D-C Network structure diagram.

Firstly, spatio-temporal features are extracted from the input feature based on the P3D convolutional layer, where H, W, T, and C denote the height, width, temporal depth, and the number of channels of the feature map, respectively. In the P3D convolutional block, the spatio-temporal features are extracted by a pseudo-3D convolutional structure (consisting of a $1 * 1 * 3$ convolutional layer and a $3 * 3 * 1$ convolutional layer) to simulate $3 * 3 * 3$ convolution to achieve the purpose of reducing the number of parameters. The formula for the number of 3D convolutional layer parameters is $(k_{\mathrm{h}} * k_{\mathrm{w}} * k_t * \mathrm{n}_{ic} + 1) * \mathrm{n}_{oc}$, where k_{h},k_{w},k_t are the sizes of 3D convolutional kernels in the three dimensions of height, width, and time, n_{ic} is the number of channels of the input feature map, and n_{oc} is the number of 3D convolutional kernels. Secondly, the channel attention (CE) mechanism is incorporated into the P3D convolutional block, which models the channel relationship of the input features, and is able to obtain the channel information weight distribution of the features, strengthen the useful channel features, and suppress the irrelevant channel features, so as to enhance the feature extraction capability of the P3D convolutional block. Finally, the output features of the P3D convolution block are obtained by fusing the output features of the $1 \times 1 \times 1$ convolution layer and the output features of the CE module using a feature fusion layer. The structure of the channel attention is schematically depicted in Figure 3.

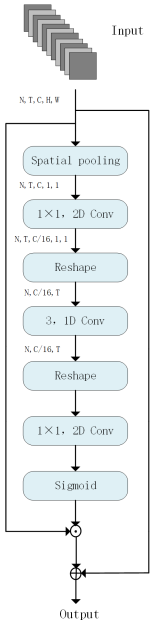


Figure 3. Channel attention (CE) structure diagram.

In the P3D-C network, N, T, C, H, W represent the batch size, the number of segments, the number of channels, and the height and width of the input image respectively. Firstly, the spatial average pooling is performed on the given input $x \in R^{[N,T,C,H,W]}$ to obtain the spatial information of the input features, and the tensor $F \in R^{[N,T,C,1,1]}$ is obtained. Secondly, the 1×1 convolution kernel is used to compress the number of channels to $1/16$ times of the original, and the tensor F_{r} is obtained. F_{r} has the receptive field of global spatio-temporal features to improve the extraction and learning ability of feature information. Then, F_{r} is reconstructed to $F_r^* \in R^{[N,C/r,T,1,1]}$, and F_r^* is processed by 1×1 convolution k_2 with kernel size 3 to obtain F_{temp}^*. In this way, temporal reasoning is performed on the information of the same channel at different times to perceive the temporal information of the channel range, and $F_R^* \in R^{[N^{C/r},T1,1]}$ is obtained by restoring the shape of Tensor using reshape. A 1×1 convolution kernel k_3 and the activation function Sigmoid are used to obtain the channel excitation matrix $M \in R^{N*T*\mathrm{C}*1*1}$. Finally, the two vectors are added to obtain the output result. The calculation process can be expressed in Eqs (3.1)–(3.6):

$$F = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X[:, :, \ldots, i, j] \tag{3.1}$$

where x denotes a four-dimensional tensor, which in CNN represents the input feature map with dimensions (Batch Size, Channels, Height, Width). H and W denote the height and width of the feature map respectively. F is the output value after global average pooling.

$$F_{\mathrm{r}} = K_1 * F \tag{3.2}$$

$$F_{temp}^* = k_2 * F_r^* \tag{3.3}$$

$$F_0 = K_3 * F_{temp} \tag{3.4}$$

where K_1, K_2, K_3 denote the transformation kernels, F_{r} denotes the feature vector after the transformation of K_1. K_3 is used to perform a weighting operation on F_{temp}, and F_0 denotes the final feature vector generated after processing by K_3.

$$M = \sigma(F_0) \tag{3.5}$$

$$out = F + F_0 \odot F \tag{3.6}$$

where *out* is the feature output from the CE module, σ denotes the sigmoid function operation, and $\odot$ denotes the feature fusion operation.

3.2. Multiscale motor attention

In the 3D convolution process, target features in output features are derived from input features and the convolution kernel through a local inner product operation within the receptive field. Consequently, 3D convolution only considers local information in input features during feature extraction. However, when processing video frame sequences, target features may depend not only on local feature information in input features but also on other spatio-temporal feature information, such as motion features. To address this, we designed a multi-scale motion attention (MME) mechanism. Explicit motion features are constructed by calculating feature differences between adjacent frames. This approach significantly

reduces the number of parameters and computational effort of the model compared to the optical flow method [25], which calculates luminance differences between neighboring pixels in a two-frame image.

The schematic structure of the motion attention mechanism is shown in Figure 4. By incorporating the multi-scale motion attention mechanism, our method can better capture and utilize spatio-temporal feature information in video frame sequences without adding excessive computational burdens, thereby enhancing the accuracy and efficiency of gesture recognition.

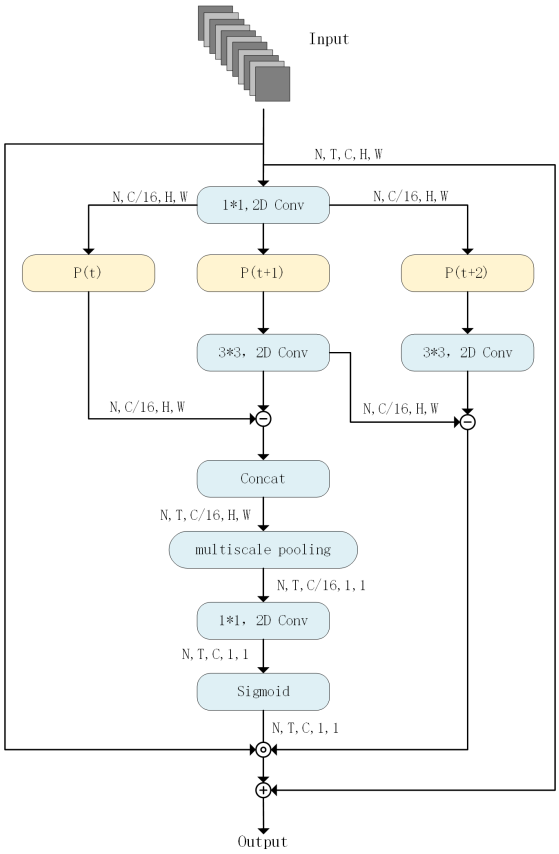


Figure 4. Multiscale motor attention (MME) structure diagram.

Firstly, the feature $x \in R^{N*T*C*H*W}$ is subjected to a 1×1 2D convolution for dimensionality reduction, resulting in $P(t)P(t+1)P(t+2) \in R^{[N,C/16,H,W]}$. Subsequently, the difference map is computed by grouping two adjacent frames of the image. The computation process is shown in Eq (3.7):

$$d_T = Conv_{3*3}(X_{T+1}) - X_T \tag{3.7}$$

where X_T denotes the input feature map at time step T, and X_{T+1} denotes the input feature map at time step $T+1$, i.e., the next frame of features immediately following X_T. $Conv3*3$ denotes a 3×3 convolution operation.

Then, the $t-1$ difference maps are stacked according to the time dimension. Due to the influence of shooting conditions, moving objects may have positional offsets between two adjacent frames. If the spatial sensing field is small, direct phase subtraction of the corresponding positions of the feature maps

can result in feature semantic mismatches and produce misclassification issues. To address this, a 3×3 spatial convolution is used to fuse neighborhood features before phase subtraction. Subsequently, the difference map is spliced into the time dimension. Since t frames of motion images can only generate an output of t-1 frames after the difference, to ensure data integrity, the t-th frame is complemented with 0 to obtain a complete D_r. Next, the D_T is inputted into the multiscale pooling (MP) layer. The structure of the multiscale pooling layer is schematically shown in Figure 5. The downsampling operation is performed through the multiscale pooling layer to obtain $D_{\mathrm{m}T} \in R^{[N,T,C/16,1,1]}$. Motor attention weight coefficients $X_{\mathrm{m}T} \in R^{[N,T,C,1,1]}$ are generated through the Softmax activation function. After obtaining the motor attention weight coefficients, they are element-wise multiplied with residual linking to the input feature $x \in R^{N*T*C*H*W}$, ultimately yielding the output feature $F \in R^{[N,T,C,1,1]}$ of the MME.

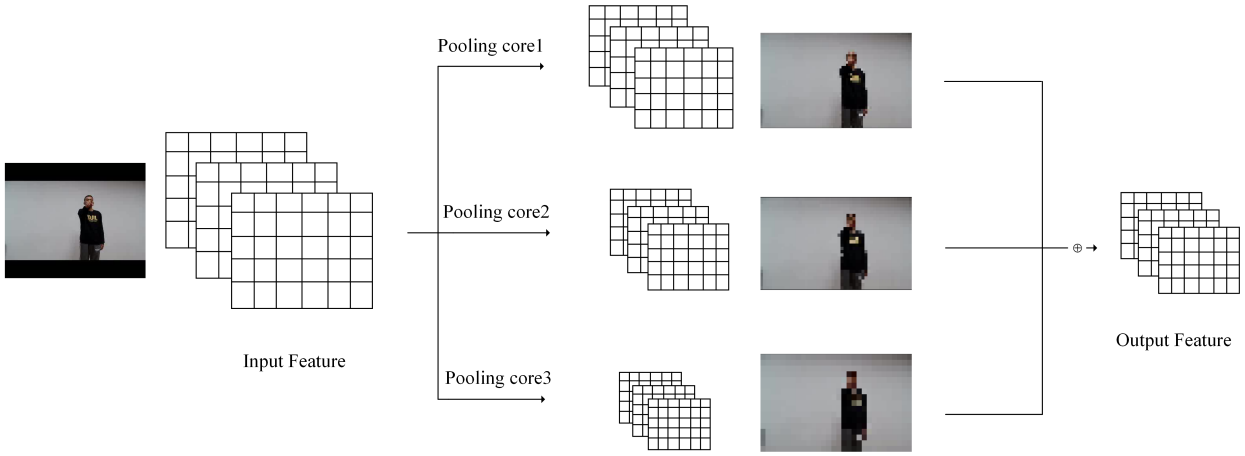


Figure 5. Multiscale Pooling (MP) layer structure diagram.

The multi-scale pooling layer structure consists of the largest pooling layers with pooling kernel sizes of 2, 4, and 6, respectively. The multi-scale pooling layer structure enables the features to be compressed from multi-dimensions and the pooled features of different scales to be extracted, which makes the network able to learn the feature information under different scales. The MME mechanism efficiently captures key information about action changes by comparing feature differences between consecutive video frames. Firstly, feature dimensionality reduction is performed, and then the difference maps between neighboring frames are computed and spliced in the time dimension to ensure data integrity. To obtain comprehensive time-series motion features, MME not only targets a single pair of adjacent frames but also applies a multi-scale pooling operation across the entire video sequence to analyze multiple pairs of consecutive frames so as to extract action information at different granularities. Finally, the local temporal features are integrated to construct a complete time-series motion feature representation so that the model can focus on the continuity and coherence of the significant action change points and the overall action process at the same time and effectively analyze and understand the motion pattern of the whole time-series. Meanwhile, the multi-scale pooling layer structure reduces the size of the feature maps in the MME, which reduces the amount of computation and parameter counts in the model.

4. Experiments

To validate the effectiveness and feasibility of the MME-P3D model in dynamic gesture recognition tasks and to further assess its complexity, we conducted experiments on both a self-constructed dataset (S-MGD) and a public dataset (ChaLearn 2013). The hardware environment for this experiment includes an NVIDIA Tesla V100 16 G graphics card, an Intel (R) Xeon (R) Gold 5218R 10-core CPU, and 29 GB of DDR4 RAM. The software platform consists of the Ubuntu 20.02-LTS operating system, Python version 3.7.10, Tensorflow version 2.27.0-GPU, CUDA version 10.1.105, and cuDNN version 7.6.4.

4.1. Dataset

4.1.1. Chalearn 2013 dataset

The Chalearn 2013 dataset is a large-scale, multimodal dynamic gesture dataset consisting of 20 Italian Sign Language gestures performed by 27 participants. The specific types of gestures can be observed in Figure 6. To facilitate analysis and evaluation, the dataset is divided into three subsets: the training set, the validation set, and the test set, with a distribution ratio of 7:2:1. Each gesture sample in the dataset includes color data, depth data, mask data, and skeletal joint point data. For our study, which focuses on vision-based gesture recognition, we exclusively utilized the color (RGB) data from the Chalearn 2013 dataset. In this particular experiment, we randomly selected six categories of sign language gesture movements from the training set's color data, extracting 200 movements for each gesture as experimental samples. These samples are then compiled and utilized for subsequent model training and performance evaluation.

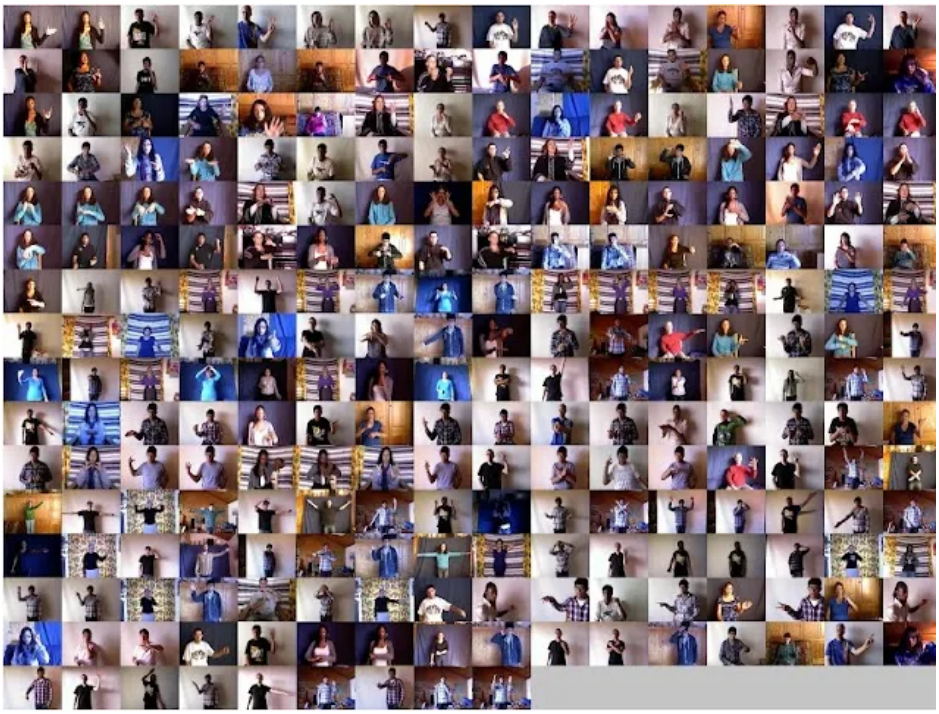


Figure 6. Chalearn partial gesture samples in the dataset.

4.1.2. Self-constructed dataset

Addressing the challenges in the field of dynamic gesture recognition and the limitations of existing publicly available datasets, we constructed a small-scale meeting gesture dataset, S-MGD (Self-established Meeting Gesture Dataset), from practical application scenarios. The dataset simulates real meeting scenarios for filming and encompasses five common gestures for controlling presentation software: capture, clicking, rotate, translation, and zoom.

1) Data collection and labelling

The S-MGD dataset employs a monocular RGB camera to record gesture instances, with the gestures of ten distinct demonstrators being captured across five action categories under varying background conditions, distances, and angles. Prior to capturing each demonstrator's gestures, the filmmaker meticulously examined the recording environment, shooting angle, and quality of sample acquisition. Following this thorough inspection, the acquisition of gesture data samples commenced. To streamline the process, enhance efficiency, and facilitate operator handling, demonstrators were instructed to perform the designated gesture actions continuously and uniformly for 90 seconds in accordance with the acquisition personnel's guidelines. They were required to pause for 5 seconds before transitioning to different gesture types, thereby easing annotation and cleaning tasks in subsequent stages. Each demonstrator was tasked with recording gestures across six scenes, encompassing three diverse light intensity levels and two alternative background environments.

Post the collection of gesture movement samples, the annotation process ensued. The essential step involved identifying the complete action clips of each gesture from every recorded video and categorically labeling them with their corresponding gesture class. Any gestures that did not align with the predefined set of five controlling presentation categories were systematically labeled as 'no background' gesture category. The comprehensive data specification for the S-MGD dataset is presented in Table 1.

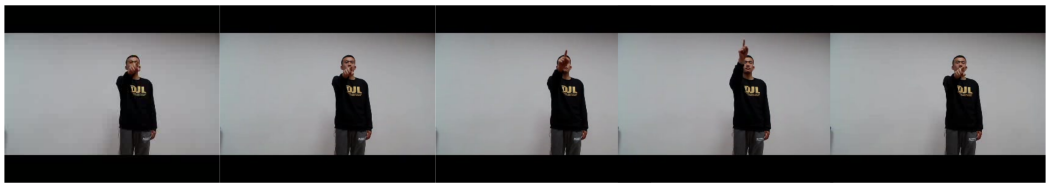
Table 1. Overview of S-MGD data sets.

Item	Data specification
Modalitites	RGB
Total number of videos	2071
Total number of freams	64,317
Number of classes	5
Number of actors	10
Avg.duration of videos	18
Avg.number of videos per class	364

To maintain consistency with the Chalearn 2013 dataset, we have standardized the image dimensions to 112×112 pixels. When the original image scale is smaller than the cropping scale, a bilinear interpolation up-sampling technique is employed for enhancement, thereby improving image clarity and ensuring effective gesture recognition classification. A selection of gesture samples is depicted in Figure 7.

Furthermore, the data length distribution of S-MGD predominantly concentrates within the range of 5–40 frames, accounting for 80.8% of the entire gesture dataset, manifesting a distinct concentra-

tion pattern. Unlike publicly available gesture datasets that typically encompass a limited number of samples with extensive variability in sample lengths, S-MGD uniquely focuses on the variability and diversity of gesture presentation speeds through the lens of sample lengths—a dimension that has been underappreciated and underemphasized in other public gesture datasets. Not only does S-MGD exhibit an overall high degree of sample length variability, but it also showcases rich variability across different gesture categories. For instance, rotation gestures primarily span between 16 and 32 frames in length, constituting 70.9% of the total samples, with only a sparse number exceeding 32 frames. Conversely, zoom gestures mostly fall within the range of 32–48 frames, representing 61.4% of the aggregate samples, with very few instances surpassing 48 frames in duration. The temporal duration of gestures varies significantly among categories, and correspondingly, so does the sample length variability within the dataset, which aligns more closely with real-world scenarios.

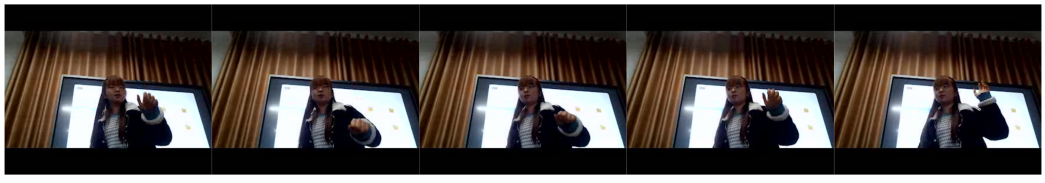


(a) Clicking

(b) Translation

(c) Zoom

(d) Rotate



(e) Capture

Figure 7. Some gesture samples in the conference gesture dataset S-MGD.

4.2. Data preprocessing

During the experiment, we processed the Chalearn 2013 dataset to maintain consistent row and column scales with the 20bn-Jester dataset (normalized to 112*112px). When the image scale was smaller than the cropping scale, we utilized bilinear interpolation up-sampling to enhance image clarity and ensure effective gesture recognition classification. To evaluate model performance, we randomly shuffled all samples and divided the training and test sets of the Chalearn 2013 and S-MGD datasets in a 4:1 ratio. This data partitioning strategy ensures independence between the training and test sets, rendering our experimental results more reliable.

4.3. Parallelism analysis

This section first assesses the parallel computing capability of the MME-P3D algorithmic model and compares it with the RNN-Model network [26] and the C3D [27] network in a comparative experiment. Among these, the RNN-Model is a neural network with a recurrent structure that updates its parameters by minimizing the difference between predicted and true results. The C3D network employs 3D convolutional kernels for convolution operations, effectively capturing spatio-temporal information in videos. The network structure of MME-P3D will not be discussed further in this context. In our experiments, we set the batch size to 8 and measured the parallel computing performance of each model using the inference time per batch.

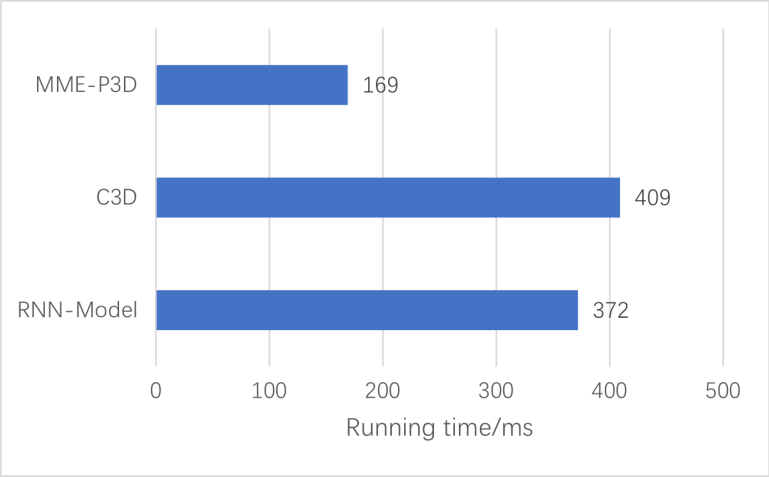


Figure 8. Model running speed comparison.

The results of comparing the running speeds of the three algorithmic models in the same experimental environment are presented in Figure 8. Due to the serialized computation of the recurrent neural network model RNN-Model, i.e., each moment's computation requires waiting for the result of the previous moment's computation, its parallel computing capability is poor, necessitating 372 ms to complete training per batch. In contrast, the C3D network model requires a training time of 409 ms, which is even less efficient due to the fact that the C3D network uses three-dimensional convolution, resulting in a larger number of parameters in the model. The MME-P3D algorithm model proposed in this paper only requires 169 ms to complete the operation, with a time cost of approximately 42% of the C3D network and 45% of the RNN model, representing a significant advantage in terms of operational

speed. This improvement is mainly attributed to MME-P3D's adoption of P3D convolution to simulate 3D convolution for spatio-temporal feature extraction and learning, significantly reducing the model's complexity. Experimental results demonstrate that the MME-P3D network framework possesses good parallel computing ability, which is crucial for enhancing the real-time performance and practicality of gesture recognition tasks.

4.4. Ablation studies

To investigate the effect of the number of P3D blocks (N) on the number of parameters, the number of computations (FLOPs), and the gesture recognition accuracy of the MME-P3D model, we conducted ablation experiments using the conference gesture dataset S-MGD. In these experiments, we set different numbers of blocks N (N = 2, 4, 6, and 8) to explore the relationship between model size and accuracy.

Table 2 and Figure 9 depict the relationship between the number of blocks and the number of model parameters and computations. For all experiments, we initialized weights using the He normal distribution method and trained them using the stochastic gradient descent (SGD) optimization algorithm [22] with a momentum parameter of 0.9. We set the initial learning rate parameter to 0.001 and performed a total of 30 training epochs.

Table 2. The effect of the number of blocks on model size.

Model	Number of blocks	Parameter quantity/M	FLOPs/G
MME-P3D	N = 2	28.67	61.05
	N = 4	33.93	73.67
	N = 6	39.19	86.29
	N = 8	44.45	98.91

From the data in Table 1, it can be observed that both the number of parameters and computational requirements of the MME-P3D model increase as the number of blocks increases. Specifically, when the block count is raised from N = 2 to N = 4, the parameter count in the MME-P3D model grows from 28.67 to 33.93 M, a rise of approximately 5.26 M, while the computational load (FLOPs) escalates from 61.05 to 73.76 G, an increment of about 12.71 G. It is worth noting that despite variations in the number of block modules, the changes in the parameter count and computational demand of the MME-P3D model remain relatively modest. The P3D model exhibits a smaller alteration in these aspects due to its use of a pseudo-3D convolutional kernel instead of a conventional 3D convolutional kernel, which leads to fewer parameters per block. Therefore, even if the number of blocks expands, it does not have a disproportionately large effect on the overall size of the MME-P3D network.

The experimental results demonstrate that by adjusting the quantity of P3D blocks, one can effectively manage the parameter count and computational requirements of the MME-P3D model, thereby optimizing its complexity and efficiency while maintaining recognition performance.

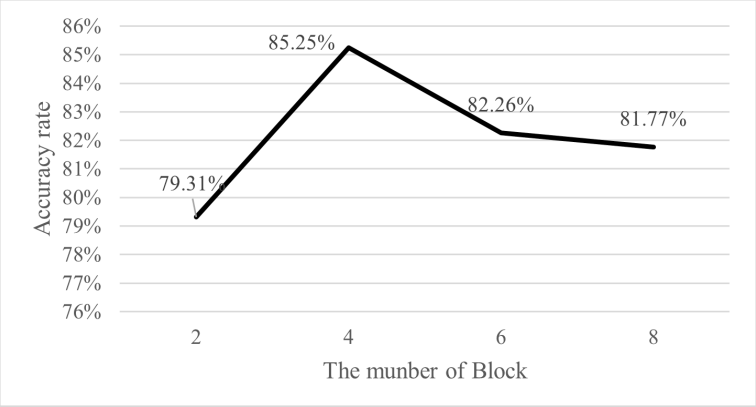


Figure 9. Change of test loss value based on S-MGD dataset.

The relationship between the number of P3D blocks and the recognition accuracy of the MME-P3D model in the S-MGD dataset is depicted in Figure 9. As the number of blocks increases, the accuracy of the model initially rises and then falls, exhibiting an inverted V-shaped trend. When N = 4, the accuracy of the model reaches its maximum value of 85.25%. Consequently, in this paper's design, we set the number of P3D blocks for the MME-P3D network model to 4.

To further validate the role of the motion attention module ME in the algorithmic model, we conducted performance analyses on both networks: MME-P3D and P3D-only (i.e., without the multiscale attention mechanism MME). The variation in test accuracy and loss values of these two models on the S-MGD dataset is shown in Figures 10 and 11.

In the early stages of training, both models exhibit faster optimization, with a significant increase in accuracy and a substantial decrease in loss values. After approximately 12 epochs, the convergence of the models begins to slow down, and changes in accuracy and loss values level off. At this stage, the test accuracy of the P3D network essentially saturates and no longer changes significantly, while the test accuracy of the MME-P3D network still experiences a small increase, reaching a steady state after about 25 epochs, which is notably better than that of the P3D network. These results suggest that the multi-scale motion attention MME can more effectively extract relevant features of gesture movement, thus positively impacting the overall recognition performance.

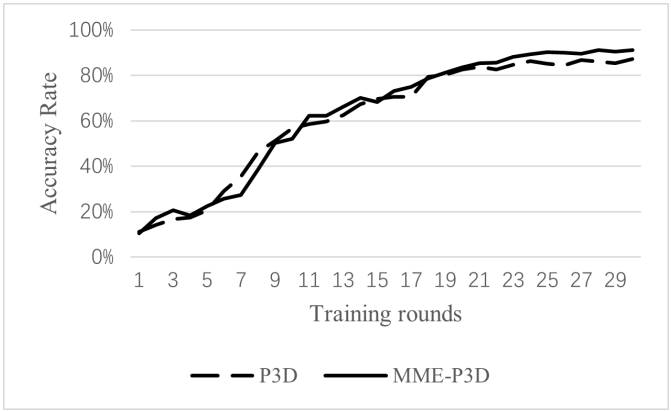


Figure 10. Changes in test accuracy based on the S-MGD dataset.

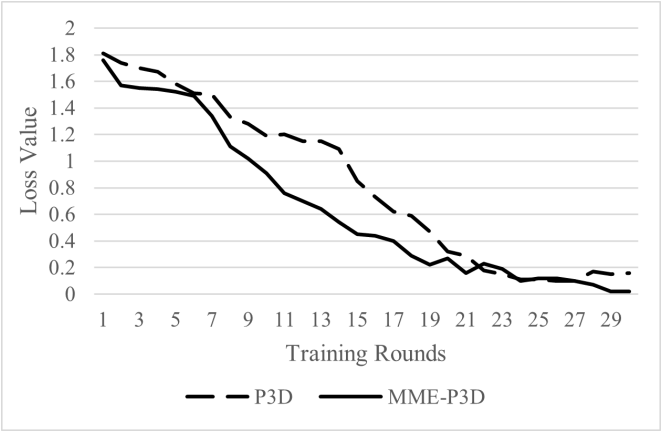


Figure 11. Changes in test loss values based on the S-MGD dataset.

From the above experimental analysis, it is evident that the multi-scale motion attention MME effectively captures and models motion feature information of hand gestures throughout, significantly improving the model's attention to effective motion information features. This enhancement further boosts the neural network's overall ability to identify the dynamics of hand gestures. By employing the P3D-C network instead of a traditional 3D convolutional network, training parameters and the number of operations can be reduced, thereby increasing the model's running speed. During the design process of the MME-P3D framework, the multi-scale motion attention MME and the P3D convolution complement each other, jointly achieving network parameter compression and improved gesture recognition performance.

4.5. Comparative experimental analysis

To validate the proposed MME-P3D algorithm model, we conducted comparative experiments on the self-built dataset S-MGD and the open dataset Chalearn 2013. We compare our model with traditional C3D networks, representative Moblienet [29] of separable convolutional networks, representative I3D [30] of short-duration 3D networks, representative 3DResnet [31] of residual networks, an online lightweight framework from the literature [21], and the MME-P3D network model proposed in this paper. During the experiment, we initialize weights using the He normal distribution method and adopt the stochastic gradient descent optimization algorithm with a momentum parameter of 0.9. We set the initial learning rate parameter to 0.001 and updated it through the cosine decay function. We employ the cross-entropy loss function, conducting the experiment for a total of 30 iteration cycles. Relevant training parameters such as batch size, learning rate, number of iterations, and weight decay are maintained consistently across different experimental methods.

The C3D network is an optimized 3D neural network that employs the concept of transfer learning to introduce VGG network parameters into a 3D convolutional network, enabling simultaneous learning of temporal and spatial features. The Moblienet network utilizes separable convolution to construct a lightweight deep neural network, achieving a balance between accuracy and network scale for excellent recognition performance. The I3D network, as a representative of long-short-term 3D networks, enhances the algorithm model's ability to extract spatio-temporal features by expanding convolution kernels. Meanwhile, the 3DResnet network leverages 3D convolutional kernels and residual struc-

tures to model and recognize hand action information. Literature [21] introduces a motion detection network, MotionNet, to determine the current presence or absence of gestures in the original video stream. MotionNet serves as a representative of the spatio-temporal network architecture that utilizes spatio-temporal convolution to capture information about the movement changes in the video with the aim of improving the performance of the action recognition task.

During testing, we segment a video clip into sequence frames and input them into the trained network model. Through forward propagation, the probability score of the gesture action category is output, and the highest probability score is selected as the prediction result, as demonstrated in Eq (4.1):

$$q = \frac{ts}{ts + fs} \tag{4.1}$$

where ts represents the number of gesture action samples correctly recognised by the model in this paper, and fs represents the number of gesture action samples incorrectly recognised by the MME-P3D model, and q represents the probability of successful gesture recognition, i.e., the accuracy rate of the model.

In this study, we employ the LOSO (Leave-One-Subject-Out) cross-validation method for our experiments. In this process, we compare the proposed MME-P3D gesture recognition algorithm with existing mainstream gesture recognition algorithms, encompassing both manual feature description-based methods and deep learning approaches. Figures 12–15 respectively display the comparison curves of loss value and accuracy of various gesture recognition algorithms during training on S-MGD and Chalearn datasets, providing an intuitive evaluation of different algorithms' performance in dynamic gesture recognition tasks.

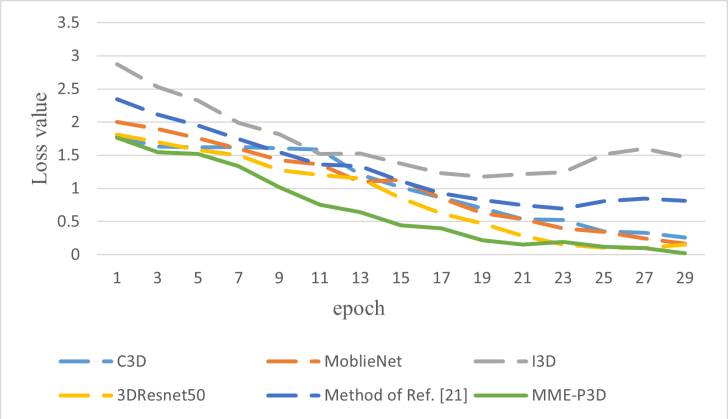


Figure 12. Loss value map on S-MGD dataset.

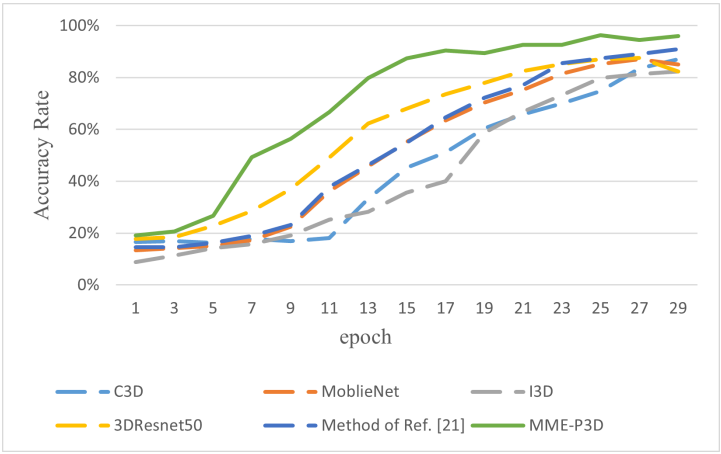


Figure 13. Accuracy variation map on S-MGD dataset.

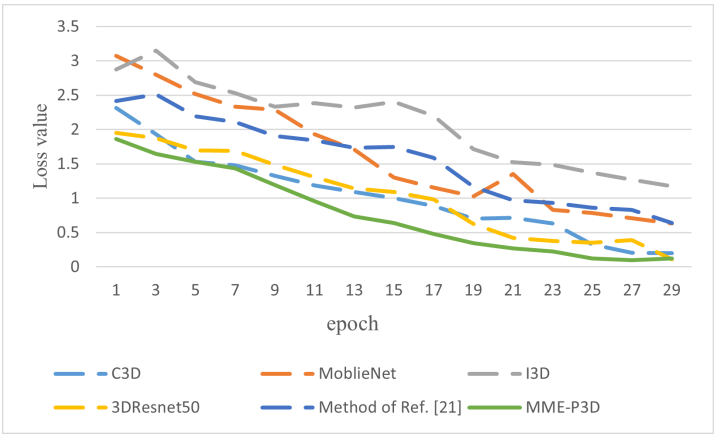


Figure 14. Loss value map on Chalearn dataset.

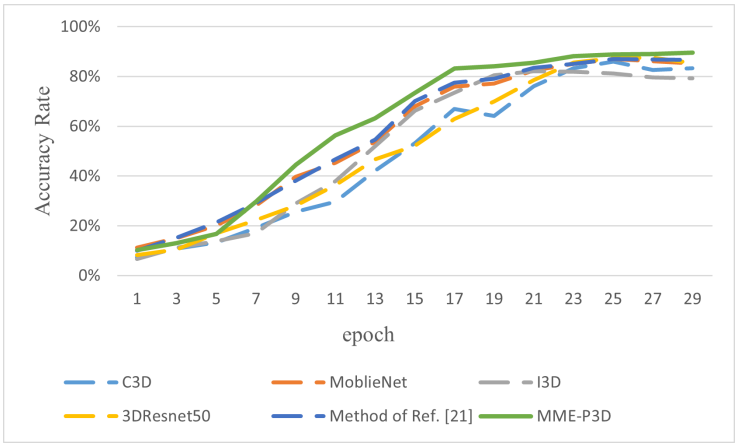


Figure 15. Accuracy variation map on Chalearn dataset.

From Figures 12–15, it is evident that the curves of the proposed MME-P3D algorithm exhibit similarities with existing mainstream gesture recognition algorithms during the first few rounds of training. However, due to the use of the P3D module in MME-P3D, the number of parameters to be learned is relatively small, which reduces computation and parameter count. Consequently, the MME-P3D model demonstrates a significant improvement in accuracy from round 5, while other models do not show substantial improvements until round 9. Additionally, the ME motion attention module in the MME-P3D model extracts motion features at multiple scales in gesture actions, causing the loss value to decrease rapidly after 5 epochs. The MME-P3D model also exhibits a smaller loss value during training compared to other gesture recognition algorithms, indicating its stronger feature extraction capability for motion information during training. In Figure 16, we can see 2 sets of results for the MME-P3D algorithm for predictive gesture recognition on video images captured in real time from the test set samples in the S-MGD dataset, for zoom and translation.

(a) Zoom

(b) Translation

Figure 16. Selected results of predictive gesture recognition on video images captured in real time from the test set samples in the S-MGD dataset.

According to the findings presented in Table 3, a comparison of accuracy was conducted between the MME-P3D network and other gesture recognition networks using the S-MGD dataset. In order to ensure the validity and accuracy of the experimental results, we compared each method under its optimal parameter settings with the aim of demonstrating the performance of each method under its optimal operating conditions. Among them, the C3D network, because it needs to capture action changes and motion features through a large spatio-temporal receptive field, was set to 32 input frames in the experiment to achieve the best performance. In contrast to 3D convolutional networks such as I3D and 3DResnet, the MME-P3D algorithm model, which employs pseudo-3D convolution as its framework, demonstrates evident advantages in computational efficiency and parameter reduction. Specifically, the MME-P3D algorithm model achieves up to an 82% and 83% reduction in calculations and parameters, respectively, significantly enhancing operational efficiency. Furthermore, with regard to recognition accuracy, the MME-P3D algorithm, incorporating multi-scale motion attention (MME), effectively extracts action features from gesture motions at various scales, facilitating improved identification and

extraction of feature information in gestures. Consequently, the MME-P3D algorithm attains an accuracy improvement of 2.91% and 5.8% compared to the lightweight two-stage framework of literature [21] and the lightweight MobileNet network. It is noteworthy that the lower accuracy observed in the 3DResnet network can be attributed to its large number of model parameters, coupled with the relatively small size of the S-MGD dataset, leading to model underfitting.

Table 3. Accuracy comparison results on S-MGD dataset.

Methods	Input frame number	Resolution	Accuracy (%)	Parameter quantity/M	FLOPs/G
C3D	32	112*112	93.37	189.11	237.68
MoblieNet	16	112*112	85.24	53.72	88.04
I3D	16	112*112	90.65	110.52	132.55
3DResnet50	16	112*112	82.27	123.87	455.23
Method of Ref. [21]	16	112*112	88.21	42.33	79.56
MME-P3D	16	112*112	**91.12**	**33.93**	**76.37**

Note: Bold font is the best value for each column.

Table 4. Accuracy comparison results on Chalearn 2013 dataset.

Methods	Input frame number	Resolution	Accuracy (%)	Parameter quantity/M	FLOPs/G
C3D	32	112*112	85.99	189.11	237.68
MoblieNet	16	112*112	74.23	53.72	88.04
I3D	16	112*112	81.16	110.52	132.55
3DResnet50	16	112*112	84.92	123.87	455.23
Method of Ref. [21]	16	112*112	79.48	42.33	79.56
MME-P3D	16	112*112	**83.06**	**33.93**	**76.37**

Note:Bold font is the best value for each column.

To further validate the effectiveness of the proposed algorithm, a series of comparative experiments were conducted on the publicly available Chalearn dataset. The results of these experiments are presented in Table 4. On the Chalearn dataset, the 3DResnet algorithm exhibits a recognition accuracy that is 2.65% higher than that of the S-MGD dataset. Notably, the MME-P3D network model introduced in this paper achieves an impressive recognition accuracy of 83.06%, which is 3.58% and 8.83% higher than the lightweight two-stage framework of literature [21] and the lightweight Mobilenet model, respectively. Furthermore, compared to the 3D convolutional network (I3D), the MME-P3D network structure demonstrates an improvement in recognition accuracy of 1.9%. This improvement can be attributed to the fact that the I3D model only utilizes information from two dimensions of space and time, whereas the MME-P3D network not only incorporates this information but also integrates the motion features of the gesture. Consequently, the MME-P3D network exhibits a higher utilization of information, resulting in superior recognition performance. Although C3D outperforms MME-P3D in terms of accuracy, considering the demands of mobile and embedded devices for limited resources, high real-time performance, and low energy consumption, MME-P3D, with its significantly reduced number of parameters and computational requirements, as well as the innovative design of the attention mechanism, achieves higher efficiency and lower resource consumption while guaranteeing a certain recognition accuracy, which makes it highly practical and valuable for research in specific application scenarios.

5. Conclusions

This paper introduces the MME-P3D gesture recognition algorithm as a solution to address the challenges posed by complex dynamic gesture recognition network models, extensive parameter counts, and computational requirements. The paper proposes an enhancement to the network's feature extraction capability by incorporating channel attention CE with the P3D network. Additionally, to overcome the limitation of P3D convolution in capturing motion information within a limited time window, the authors introduce MME as a supplementary component to facilitate better understanding and learning of dynamic information during gesture motion. Experimental findings indicate that the MME-P3D model achieves a recognition accuracy of 91.12% on the S-MGD dataset of conference gestures, while the recognition accuracy on the Chalearn 2013 gesture dataset is 83.06%. Furthermore, the proposed model exhibits noticeable advantages in terms of parameter count and computational requirements, with reductions of up to 82% and 83%, respectively, when compared to the 3D convolutional neural network. Despite these reductions, the accuracy of our algorithm remains consistent with other dynamic gesture recognition methods and does not significantly lag behind. The research demonstrates that the proposed algorithm not only reduces the parameter count and computational burden of the model but also ensures accurate gesture recognition. This characteristic addresses the limitations of the original algorithm and renders MME-P3D more suitable for deployment on embedded and mobile devices.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was partly supported by the Natural Science Basis Research Plan in Shaanxi Province of China under Grant 2023-JC-YB-517, and the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University under Grant VR-LAB2023B08.

Conflict of interest

All of the authors declare that there is no conflict of interest regarding the publication of this article and would like to thank the anonymous referees for their valuable comments and suggestions.

References

1. Y. Zhang, J. Wang, X. Wang, H. Jing, Z. Sun, Y. Cai, Static hand gesture recognition method based on the vision transformer, *Multimedia Tools Appl.*, **82** (2023), 1–20. https://doi.org/10.1007/s11042-023-14732-3
2. T. Zhang, Application of AI-based real-time gesture recognition and embedded system in the design of English major teaching, *Wireless Netw.*, **2021** (2021), 1–13. https://doi.org/10.1007/s11276-021-02693-0

3. Y. Xue, Y. Yu, K. Yin, P. Li, S. Xie, Z. Ju, Human in-hand motion recognition based on multi-modal perception information fusion, *IEEE Sens. J.*, **22** (2022), 6793–6805. https://doi.org/10.1109/JSEN.2022.3148992

4. M. S. Amin, S. T. H. Rizvi, M. M. Hossain, A comparative review on applications of different sensors for sign language recognition, *J. Imaging*, **8** (2022), 1–48. https://doi.org/10.3390/jimaging8040098

5. Y. Zhang, Y. Huang, X. Sun, Y. Zhao, X. Guo, P. Liu, et al., Static and dynamic human arm/hand gesture capturing and recognition via multiinformation fusion of flexible strain sensors, *IEEE Sens. J.*, **20** (2020), 6450–6459. https://doi.org/10.1109/JSEN.2020.2965580

6. N. H. Dardas, N. D. Georganas, Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques, *IEEE Trans. Instrum. Meas.*, **60** (2011), 3592–3607. https://doi.org/10.1109/TIM.2011.2161140

7. B. Qiang, Y. Zhai, M. Zhou, X. Yang, B. Peng, Y. Wang, et al., SqueezeNet and fusion network-based accurate fast fully convolutional network for hand detection and gesture recognition, *IEEE Access*, **9** (2021), 77661–77674. https://doi.org/10.1109/ACCESS.2021.3079337

8. P. Barros, S. Magg, C. Weber, S. Wermter, A multichannel convolutional neural network for hand posture recognition, in *Artificial Neural Networks and Machine Learning–ICANN 2014: 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15–19, 2014. Proceedings 24*, (2014), 403–410. https://doi.org/10.1007/978-3-319-11179-7_51

9. S. Gnanapriya, K. Rahimunnisa, A hybrid deep learning model for real time hand gestures recognition, *Intell. Autom. Soft Comput.*, **36** (2023), 763–767. https://doi.org/10.32604/iasc.2023.032832

10. Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, et al., Multimodal gesture recognition based on the resc3d network, in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, (2017), 3047–3055. https://doi.org/10.1109/ICCVW.2017.360

11. D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in *2015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 4489–4497. https://doi.org/10.1109/TIP.2021.3092828

12. J. Wan, S. Escalera, G. Anbarjafari, H. Jair Escalante, X. Baró, I. Guyon, et al., Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, (2017), 3189–3197. https://doi.org/10.1109/ICCVW.2017.377

13. Z. Z. Wang, Automatic and robust hand gesture recognition by SDD features based model matching, *Appl. Intell.*, **52** (2022), 11288–11299. https://doi.org/10.1007/s10489-021-02933-y

14. Q. Gao, Y. Chen, Z. Ju, Y. Liang, Dynamic hand gesture recognition based on 3D hand pose estimation for human–robot interaction, *IEEE Sens. J.*, **18** (2021), 17421–17430. https://doi.org/10.1109/JSEN.2021.3059685

15. Y. Li, D. Zhang, J. Chen, J. Wan, D. Zhang, Y. Hu, et al., Towards domain-independent and real-time gesture recognition using mmwave signal, *IEEE Trans. Mob. Comput.*, **22** (2022), 7355–7369. https://doi.org/10.1109/TMC.2022.3207570

16. M. Wang, X. Li, S. Chen, X. Zhang, L. Ma, Y. Zhang, Learning representations by contrastive spatio-temporal clustering for skeleton-based action recognition, *IEEE Trans. Multimedia*, **2023** (2023), 1–14. https://doi.org/10.1109/TMM.2023.3307933

17. C. Pang, X. Gao, Z. Chen, L. Lyu, Self-adaptive graph with nonlocal attention network for skeleton-based action recognition, *IEEE Trans. Neural Networks Learn. Syst.*, **2023** (2023), 1–13. https://doi.org/10.1109/TNNLS.2023.3298950

18. P. Geng, X. Lu, C. Hu, H. Liu, L. Lyu, Focusing fine-grained action by self-attention-enhanced graph neural networks with contrastive learning, *IEEE Trans. Circuits Syst. Video Technol.*, **33** (2023), 4754–4768. https://doi.org/10.1109/TCSVT.2023.3248782

19. W. Song, T. Chu, S. Li, N. Li, A. Hao, H. Qin, Joints-centered spatial-temporal features fused skeleton convolution network for action recognition, *IEEE Trans. Multimedia*, **2023** (2023), 1–15. https://doi.org/10.1109/TMM.2023.3324835

20. C. Pang, X. Lu, L. Lyu, Skeleton-based action recognition through contrasting two-stream spatial-temporal networks, *IEEE Trans. Multimedia*, **25** (2023), 8699–8711. https://doi.org/10.1109/TMM.2023.3239751

21. C. Xu, X. Wu, M. Wang, F. Qiu, Y. Liu, J. Ren, Improving dynamic gesture recognition in untrimmed videos by an online lightweight framework and a new gesture dataset ZJUGesture, *Neurocomputing*, **523** (2023), 58–68. https://doi.org/10.1016/j.neucom.2022.12.022

22. Z. Qiu, T. Yao, T. Mei, Learning spatio-temporal representation with pseudo-3D residual networks, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 5533–5541. https://doi.org/10.1109/ICCV.2017.590

23. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 6450–6459. https://doi.org/10.1109/CVPR.2018.00675

24. S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 318–335. https://doi.org/10.1007/978-3-030-01267-0_19

25. Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, Flowformer: A transformer architecture for optical flow, in *European Conference on Computer Vision*, (2022), 668–685. https://doi.org/10.1007/978-3-031-19790-1_40

26. J. Wang, X. Li, J. Li, Q. Sun, H. Wang, NGCU: A new RNN model for time-series data prediction, *Big Data Res.*, **27** (2022), 100296. https://doi.org/10.1016/j.bdr.2021.100296

27. T. Huynh-The, C. H. Hua, N. A. Tu, D. S. Kim, Learning 3D spatiotemporal gait feature by convolutional network for person identification, *Neurocomputing*, **397** (2020), 192–202. https://doi.org/10.1016/j.neucom.2020.02.048

28. B. Zhou, C. Han, T. Guo, Convergence of stochastic gradient descent in deep neural network, *Acta Math. Appl. Sin.*, **37** (2021), 126–136. https://doi.org/10.1007/s10255-021-0991-2

29. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

30. J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, preprint, arXiv:1705.07750. https://doi.org/10.48550/arXiv.1705.07750

31. H. Kataoka, T. Wakamiya, K. Hara, Y. Satoh, Would mega-scale datasets further enhance spatiotemporal 3D CNNs, preprint, arXiv:2004.04968. https://doi.org/10.48550/arXiv.2004.04968

AIMS Press