Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061396

ARTICLE

Video Action Recognition Method Based on Personalized Federated Learning and Spatiotemporal Features

Rongsen Wu[1], Jie Xu[1], Yuhang Zhang[1], Changming Zhao[2,*], Yiweng Xie[3], Zelei Wu[1], Yunji Li[2], Jinhong Guo[4] and Shiyang Tang[5,6]

[1]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[2]School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China
[3]Shanghai Key Lab of Intelligent Information Processing, School of CS, Fudan University, Shanghai, 200433, China
[4]School of Sensing Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[5]School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, 2052, Australia
[6]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK

*Corresponding Author: Changming Zhao. Email: zcm84@cuit.edu.cn

Received: 29 December 2024; Accepted: 07 March 2025; Published: 19 May 2025

ABSTRACT: With the rapid development of artificial intelligence and Internet of Things technologies, video action recognition technology is widely applied in various scenarios, such as personal life and industrial production. However, while enjoying the convenience brought by this technology, it is crucial to effectively protect the privacy of users' video data. Therefore, this paper proposes a video action recognition method based on personalized federated learning and spatiotemporal features. Under the framework of federated learning, a video action recognition method leveraging spatiotemporal features is designed. For the local spatiotemporal features of the video, a new differential information extraction scheme is proposed to extract differential features with a single RGB frame as the center, and a spatial-temporal module based on local information is designed to improve the effectiveness of local feature extraction; for the global temporal features, a method of extracting action rhythm features using differential technology is proposed, and a time module based on global information is designed. Different translational strides are used in the module to obtain bidirectional differential features under different action rhythms. Additionally, to address user data privacy issues, the method divides model parameters into local private parameters and public parameters based on the structure of the video action recognition model. This approach enhances model training performance and ensures the security of video data. The experimental results show that under personalized federated learning conditions, an average accuracy of 97.792% was achieved on the UCF-101 dataset, which is non-independent and identically distributed (non-IID). This research provides technical support for privacy protection in video action recognition.

KEYWORDS: Video action recognition; personalized federated learning; spatiotemporal features; data privacy

1 Introduction

In recent years, video action recognition technology has made significant progress and is widely applied in fields such as intelligent surveillance, human-computer interaction, and sports analysis. Due to breakthroughs in deep learning algorithms, especially the widespread use of models like CNN and LSTM, the accuracy and robustness of video action recognition have been greatly improved [1,2]. However, with the further development of these technologies, data privacy and security issues related to video

action recognition have become increasingly prominent, emerging as key factors that constrain its broader adoption [3,4]. For instance, in a home environment, a surveillance camera must continuously record images of household members, which inherently involves capturing sensitive data related to the user's daily life and privacy. Therefore, we need to consider how to ensure user privacy in data collection and processing. On the one hand, we must ensure that data processing and application tasks are performed locally, with no possibility of uploading the data to a server. This requires us to consider security in the design of data processing systems to avoid data leaks and abuse. On the other hand, we need to prevent the dissemination and storage of data across distances from causing security risks.

Federated learning algorithms can effectively solve the privacy issue of user data in video monitoring. Under the framework of federated learning, the model can use scarce data locally to complete training, and the models of each user are aggregated through different methods and strategies on the central server. The data do not need to be transmitted to the central server, ensuring the security of data privacy. Utilizing a federated learning framework for video action recognition can significantly enhance the security of user privacy [5–9].

This paper integrates federated learning, fully considering user data security and privacy, and proposes a video action recognition method based on personalized federated learning and spatiotemporal features. Under the framework of federated learning, this method designs a video action model that extracts spatiotemporal features using differential methods. The model embeds a spatiotemporal module based on local information and a temporal module based on global information within a residual network structure. These two modules respectively use RGB differences and feature differences to extract local spatiotemporal features and global temporal features, thereby improving model efficiency while ensuring recognition performance. The main contributions of this paper are as follows:

(1) To address the needs of privacy protection and data security in video surveillance, this paper combines federated learning with video action recognition models to propose a method based on personalized federated learning and spatiotemporal features.
(2) Regarding the spatiotemporal features of video segments, this paper proposes a spatial-temporal module based on local information, which uses a new differential information extraction method to provide complementary spatial static information with temporal features.
(3) Regarding the time-based features of complete videos, this paper proposes a time module based on global information that utilizes differential information of local features to extract action rhythm features, thereby improving the extraction effect of time-based features.

2 Related Work

In the field of personalized federated learning for video action recognition, scholars have proposed various innovative solutions. Zhao et al. [10] proposed an activity recognition system that uses semi-supervised federated learning, where clients use unlabeled local data to learn general representations through long short-term memory autoencoders, and the cloud server uses labeled data with a Softmax classifier for supervised learning. Experimental results show that their proposed system achieves higher accuracy than centralized systems and semi-supervised federated learning with data augmentation, and its accuracy is comparable to that of supervised federated learning systems. Shome et al. [11] proposed a federated learning framework for facial expression recognition that uses a small amount of labeled private facial expression data to train local models in each training round and aggregates all local model weights on the central server to obtain the global optimal model. Rehman et al. [12] proposed a general FL framework FedVSSL based on SWA for pre-training video-SSL methods in FL. This method shows strong competitiveness in action recognition tasks compared to FedAvg and centralized video SSL. Doshi et al. [13] proposed an

effective federated learning solution based on 2D CNN models for detecting distracted driver activities. This solution trains the detection model in a distributed manner while protecting privacy and reducing data communication. Tu et al. [14] proposed a federated few-shot learning framework FedFSLAR, which collaboratively learns classification models from multiple FL clients using a small number of labeled video samples to recognize unknown actions.

3 Video Action Recognition Method Based on Personalized Federated Learning and Spatiotemporal Features

The proposed video action recognition method based on personalized federated learning and spatiotemporal features incorporates the characteristics of the video action recognition model to divide the parameters into private and public parameters. First, the complete process of the personalized federated learning method and the production method of non-independent and same distribution video action recognition datasets are given. Second, this section introduces the specific structure of the video action recognition model.

3.1 Overview

The proposed personalized federated learning scheme combines the characteristics of video action recognition models to divide private parameters and public parameters. Meanwhile, federated learning allows for training directly on edge devices, eliminating the need to transmit raw data from client devices to a central server, thereby reducing the risk of data leakage [15]. The video action recognition model is divided into three parts: input, local feature extraction, and global feature extraction. Taking the overall model segmentation number $n = 3$ as an example, $n = 8$ and $n = 16$ will be used for experimental results in subsequent experiments. In the input stage, video data are divided into three segments; for each segment, one frame of RGB is sampled first followed by taking the first two frames before and after the selected frame, respectively, and calculating the difference with the selected frame to obtain four RGB differences. In the local feature extraction stage, the differences are stacked and passed through a pooling layer to obtain initial local difference information, which is then input together with the video frame into the second stage network of ResNet for spatial feature extraction. Another copy of the initial difference information is also input into the second stage network of ResNet to extract temporal features and the spatial features are added to the temporal features to obtain the final local spatial features of each segment. In the global feature extraction stage, each segment's local features are compressed in the channel dimension and a bidirectional global difference is obtained by translating each segment's features. Through a convolutional neural network, global features are obtained, corresponding to the last three stages of ResNet with different stacking layers. Three rounds of global feature extraction are performed in total. Finally, the global features are input into a classifier to obtain the final video action recognition results.

When performing local feature extraction, the model focuses more on the static information of video data, including some key image information such as human body, color, and objects, so it is highly dependent on training data. Local features, serving as private parameters for users, are retained on the users' local devices and are not uploaded to the central server. This ensures that users' original video data and personalized features never leave their devices, thereby greatly enhancing privacy protection. Conversely, global feature extraction focuses more on dynamic information and extracts features that change with time. In personalized federated learning, local feature extraction is more suitable for training and storage at the user's local device, corresponding to the private parameters of the local model, whereas global features are better suited for aggregation at a central server, resulting in shared parameters that are saved as the public model. Fig. 1 shows the personalized federated learning-based video action recognition method.

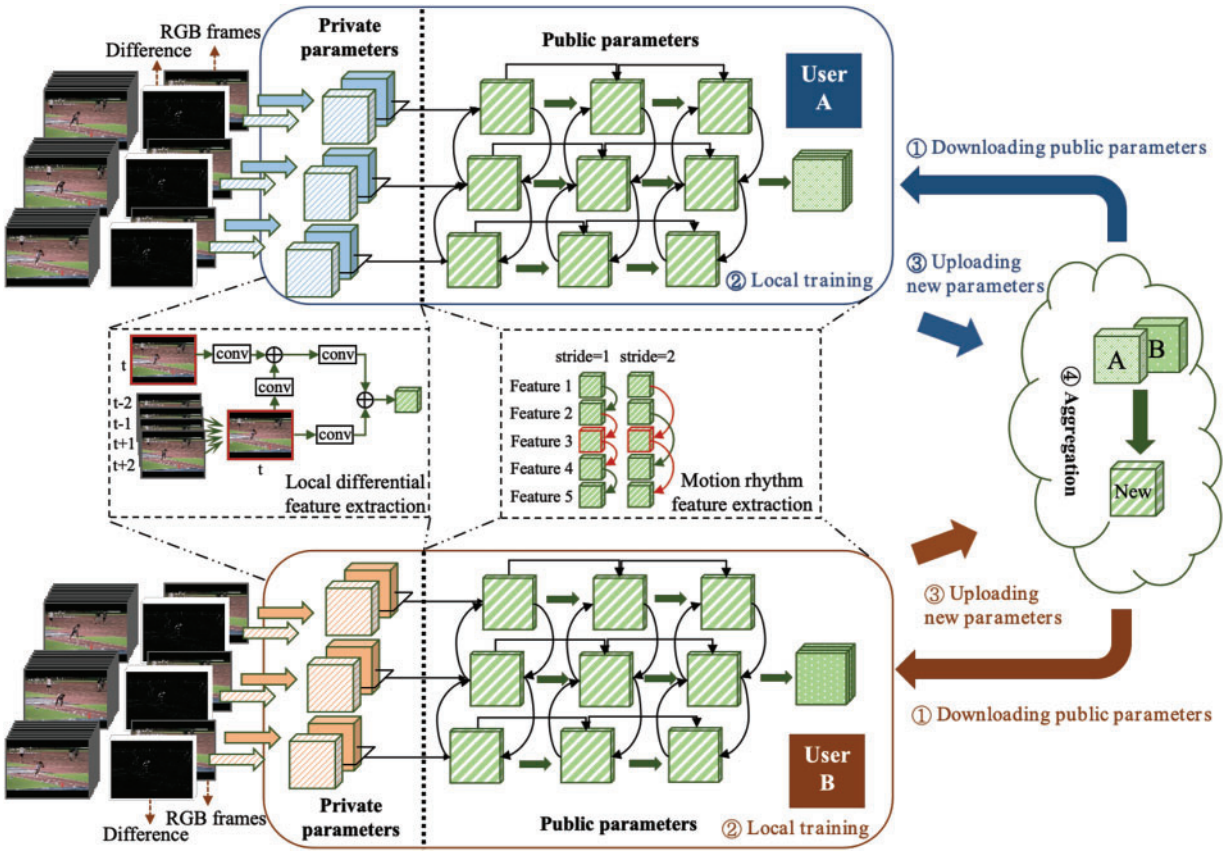

Figure 1: Diagram of the video action recognition framework based on personalized federated learning and spatiotemporal features

This paper combines the video action recognition model with the personalized federated learning mechanism. For the model in the four feature extraction stages, the parameters of the local feature extraction stage are used as personalized models for each user, while the parameters of the global feature extraction stage are used as global shared parameters. In Fig. 1, the blue rectangle represents the private parameters of user A, the red rectangle represents the private parameters of user B, and the green diagonal rectangle represents the public parameters downloaded from the central server after step ①. After backpropagation and parameter update in local training (step ②), the entire model undergoes different changes for each user. Then, the updated public parameters are uploaded to the central server through step ③ and the parameters are aggregated again by the central server in step ④ to obtain new public parameters. The process is repeated for a new round of federated learning communication.

3.1.1 Spatial-Temporal Module Based on Local Information

In recent years, some efficient methods for obtaining temporal information have been proposed. Among them, RGB difference and temporal shift methods are both simple and effective. RGB difference can simply obtain boundary and action information by performing a difference between RGB frames, whereas a temporal shift shifts the feature map in the time dimension, allowing features to overlap in time and extract dynamic information during further feature extraction.

The local spatiotemporal module proposed in this paper uses a convolutional neural network to extract features from RGB frames, obtaining local spatial features, and then extracts supplementary temporal information from the difference between multiple frames of RGB over a period of time to obtain the local spatiotemporal information of the current video segment. This module addresses the issues of traditional methods relying on complex data preprocessing and time-consuming processes when extracting short-term temporal information from local regions. It achieves a more efficient way to obtain temporal information and enhances model performance. The structure of the local module is shown in Fig. 2 and the entire module can be divided into two branches. For the input frame I_t at time t, the first branch directly inputs the raw input data into the convolution layer to extract features, obtaining the static information of the current video frame and the original spatial features of the current segment. The second branch obtains data from two frames before and after t, performs a difference operation on a total of five frames, smooths the feature in the channel dimension, passes through an average pooling layer in the planar dimension, and then adds the pooled feature to input into the convolutional network. At this time, it can obtain the supplementary temporal features of the current video segment. These features are divided into two paths: one directly inputs into the second-stage network of ResNet to extract features and the other combines with the static feature and upsamples according to the feature shape of the first branch, adding it to the first branch feature and inputting into ResNet. Finally, the two feature maps are re-scaled and added to obtain the final local spatiotemporal feature.

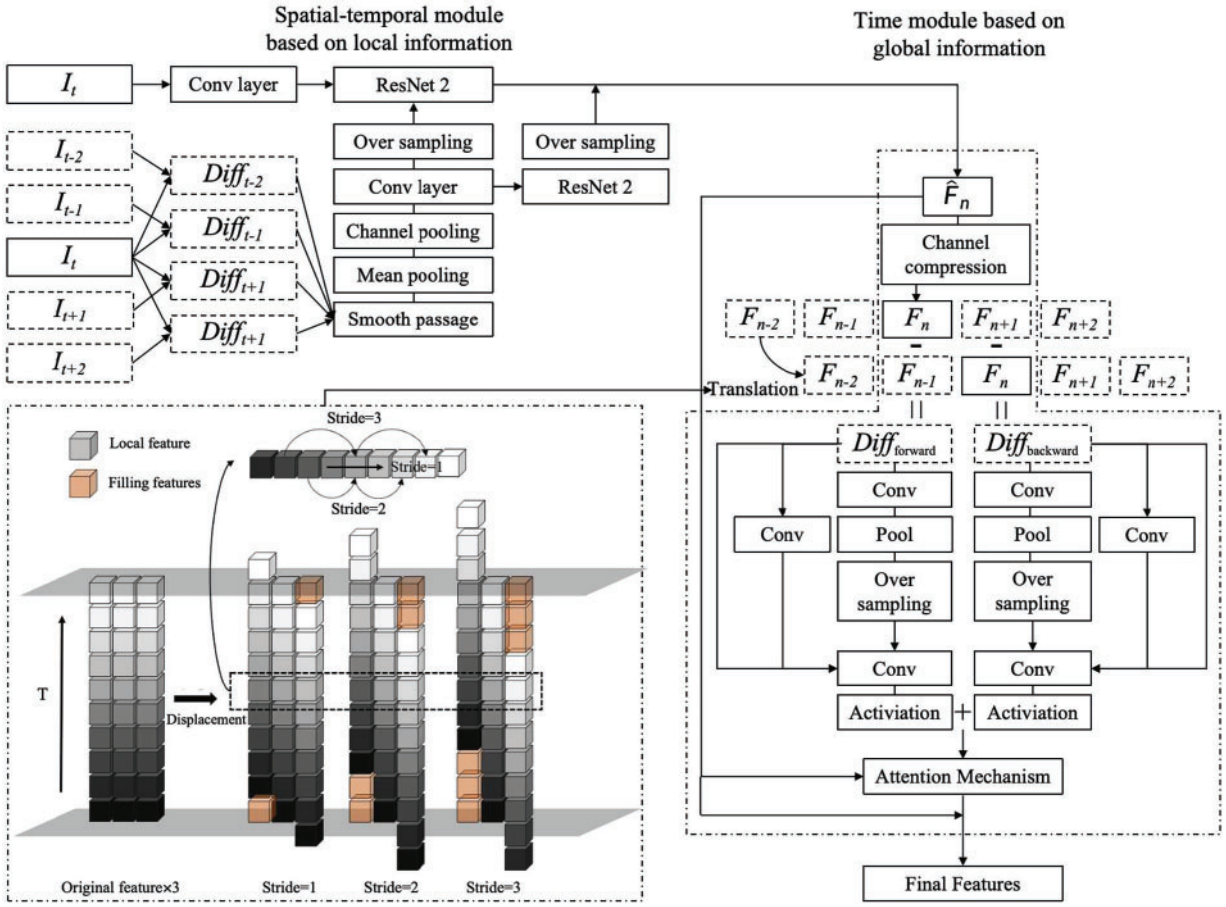

Figure 2: Spatial-temporal module based on local information and time module based on global information

Theoretically, the neighboring frames of the input frame I_t at time t are I_{t-2}, I_{t-1}, I_{t+1}, and I_{t+2}. The difference between these 4 frames and I_t is taken, with F_{diff} representing the differential features and F_{RGB} representing the static features. The calculation process of the two paths can be represented by Eqs. (1) and (2):

$$F_{diff} = \text{Conv} \sum_{i=-2,-1,1,2} \text{Avg}(I_t - I_{t-i}) \tag{1}$$

$$F_{RGB} = \text{Conv}(I_t) \tag{2}$$

In the equations, Avg represents the average pooling layer mapping, and $Conv$ represents the convolutional layer mapping. Finally, F_L represents the features of the local module, which can be expressed by Eq. (3):

$$F_L = ResNet[F_{RGB} + UpSample(F_{diff})] + UpSample[ResNet(F_{diff})] \tag{3}$$

In the equation, $UpSample$ represents the upsampling of features, and $ResNet$ represents the residual network mapping.

3.1.2 Time Module Based on Global Information

After obtaining the local spatiotemporal features of video segments, it is necessary to further acquire the temporal features between segments. Both local and global temporal features are important for action recognition. Some video actions may occur in a few moments but have a fixed order. It is necessary to extract local spatiotemporal features and further interact features across the entire temporal dimension of the video. Some actions are slow and continuous actions, requiring the model to grasp the data features of each stage.

For this reason, this paper further proposes a time module based on global information, which uses feature differencing to extract action rhythm information. The input of the module is the local spatiotemporal features of each segment. For features of different segments, differential interaction can be performed through fixed time rules to extract the time features of fixed action rhythms. In the model proposed in this paper, different time intervals essentially represent different action rhythms.

Fig. 2 illustrates the overall structure of the time module based on global information. For the local feature F_n of the nth segment, a backup is first saved and directly connected to the lower-level network, which is theoretically similar to a residual network, preventing gradient and degradation problems and accelerating propagation. Meanwhile, the original frame-level features can also be retained in the current module. Secondly, the original F_n is input into a convolutional network to achieve compression in the channel dimension, smoothing the features. This is because there is a large gap between the features of different segments and direct differencing operations may introduce a significant amount of noise, disrupting the original spatiotemporal features. Smoothing the features in the channel dimension before differencing makes the differencing features more effective. The smoothed features are also backed up for later differencing calculations. Another copy is input into the convolutional layer for feature extraction and then the difference is calculated with the backup features of other segments to obtain the difference information of different segment features. Different features participate in differencing interactions under different translation strides.

In existing video action recognition models based on action rhythm features, different action rhythm features are often extracted by sampling at different data frequencies, and different input sizes require separate network channels for feature extraction, greatly increasing the model parameter count and training time cost. The approach proposed in this paper for extracting action rhythm features directly implements differential

feature extraction at different intervals within existing local features through different translation strides, thereby controlling the model size while improving recognition performance.

For the extraction of action rhythm, as shown in the feature vector at the top of Fig. 2, different features of interaction under different displacement step sizes are marked on the vector. When the step size is small, action changes within a relatively short period of time can be obtained, which is suitable for extracting fast-paced action features. Similarly, when the step size is large, it captures slow-paced action changes. In this paper, the method of calculating differences is still used to obtain the temporal features between segments. This module addresses the issue of efficiently extracting video action rhythm features while reducing noise interference by smoothing the features before differencing.

In the specific implementation process, for the data feature F_t, displaced features are obtained through bidirectional translation. The diagrams below Fig. 2 illustrate displacement step sizes of 1, 2, and 3. On this basis, features beyond the boundaries are removed and the blank features are filled in to obtain three feature vectors that are displaced in the time dimension. Then, subtracting the three vectors yields differential information for different time spans. Blank features appearing after displacement are directly filled with null values.

Using the above method, bidirectional differential features can be obtained after bidirectional translation. Then, the features are divided into three paths: one path passes through pooling layers, convolutional layers, and upsampling layers before being passed to the next layer while the other passes through convolutional layers before being passed to the next layer; one path directly transmits the original features downward, and the three paths are added in the next layer. This approach can further enhance the robustness of the time module, making the smoothing operation on different segment features more effective. Subsequently, the features are fused deeply again through convolutional layers and activation layers, and the fused bidirectional features are added together to obtain the bidirectional differential features of the current video segment. Afterwards, the differential features are multiplied with the original features one by one, which is equivalent to treating the differential features as attention parameters of the original features. Attention mechanisms are often more effective at higher levels of network structure, so this paper applies them to the time module. Finally, the segment features with attention mechanisms are added to the original features to obtain the final features of the time module.

Using D to represent the differential function, the differential calculation process can be expressed by Eq. (4):

$$D_{n,n-1} = Conv(F_n) - Conv(F_{n-1}) \tag{4}$$

Using F to represent the smoothed features, and $D_{n,n-1}$ represents the local feature difference between the $n-1$ and n segments.

Next, using H to represent the merged features of the three paths, and H' to represent the fused features, the calculation process of the unidirectional features is shown in Eqs. (5) and (6):

$$H_{n,n-1} = D_{n,n-1} + Conv(D_{n,n-1}) + Conv[UpSample(D_{n,n-1})] \tag{5}$$

$$H'_{n,n-1} = Sigmoid[Conv(H_{n,n-1})] \tag{6}$$

Using $UpSample$ to represent upsampling, and the upsampling function is used again in the global module to unify the size of the three path features. $Sigmoid$ is the activation function used in this layer of the temporal module. Finally, using F to represent the final features output by the temporal module, as shown

in equation Eq. (7):

$$F'_n = \widehat{F}_n + \widehat{F}_n \odot \frac{1}{2}[H'_{n,n-1} + H'_{n+1,n}] \tag{7}$$

Using $\widehat{F}_n$ to represent the original local features of the nth segment, and $\odot$ represents element-wise multiplication. This means the original features are multiplied with the bidirectional features and then added, and finally the original features are added again to obtain the temporal information of the current stage.

3.2 Federated Learning of Video Action Recognition Dataset

To verify the performance of the video action recognition model under the federated learning training mechanism, a federated learning video dataset was created using the publicly available video dataset UCF-101.

As for federated learning, considering user privacy, each user trains the model using local data. In the field of machine learning, datasets often follow the Independent Identically Distributed (IID) assumption, but in the practical application scenario of federated learning, the data distribution of each user is irregular and belongs to the Non-Independent Identically Distributed (non-IID) dataset [16,17]. Existing federated learning research usually groups data based on existing public datasets, mainly in two ways. As shown in Fig. 3, taking five users as an example, each user contains three types of data, where the vertical axis represents the user number, the horizontal axis represents the number of data samples, and different colors represent different data categories. The first method in Fig. 3 directly divides the dataset into categories and assigns fixed category data to each user, with no overlap between users. Each category may have different sample sizes and limited public datasets provide different granularities of classification. Reference can be made to the large category grouping provided in the dataset for user-specific data allocation.

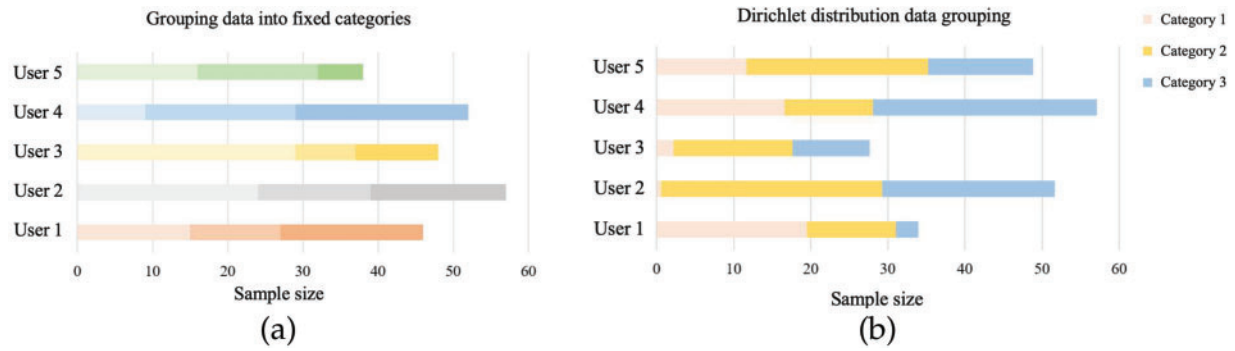

Figure 3: Example of dataset grouping using Dirichlet distribution and fixed category groups. (a) Grouping data into fixed categories; (b) Dirichlet distribution data grouping

The dataset contains N classes and it is assumed that each user's subset of data is independently dependent on the column-specific distribution parameter vector q, which satisfies the condition given in Eq. (8):

$$q_i \sim Dir(\alpha p), \quad q_i \geq 0, \quad i \in [1, N] \quad and \quad \|q\| = 1 \tag{8}$$

$Dir(*)$ represents the Dirichlet distribution, p is a prior distribution based on N, and $\alpha > 0$ is a core parameter used to control the independence of user-specific data subsets. As α approaches infinity,

the category distribution of the subsets approaches that of the original dataset. On the other hand, as α approaches 0, each user only contains one randomly assigned category.

Fig. 4 shows the Dirichlet grouping of the UCF-101 dataset under different values of α. The experiment used the UCF-101 dataset, which has 101 classes displayed on the vertical axis. The horizontal axis represents 50 users and the darkness of the colors represents the percentage of samples assigned to a particular category for a given user out of the total samples. When the colors are the same or similar across rows, it represents that the category was evenly distributed among the 50 users, resulting in each user receiving an equal proportion of samples. It can be seen that when α = 500, most categories are evenly distributed among users, whereas when α = 5, all categories are scattered and disorganized, forming a non-independent and non-identically distributed (non-IID) dataset between the subsets.

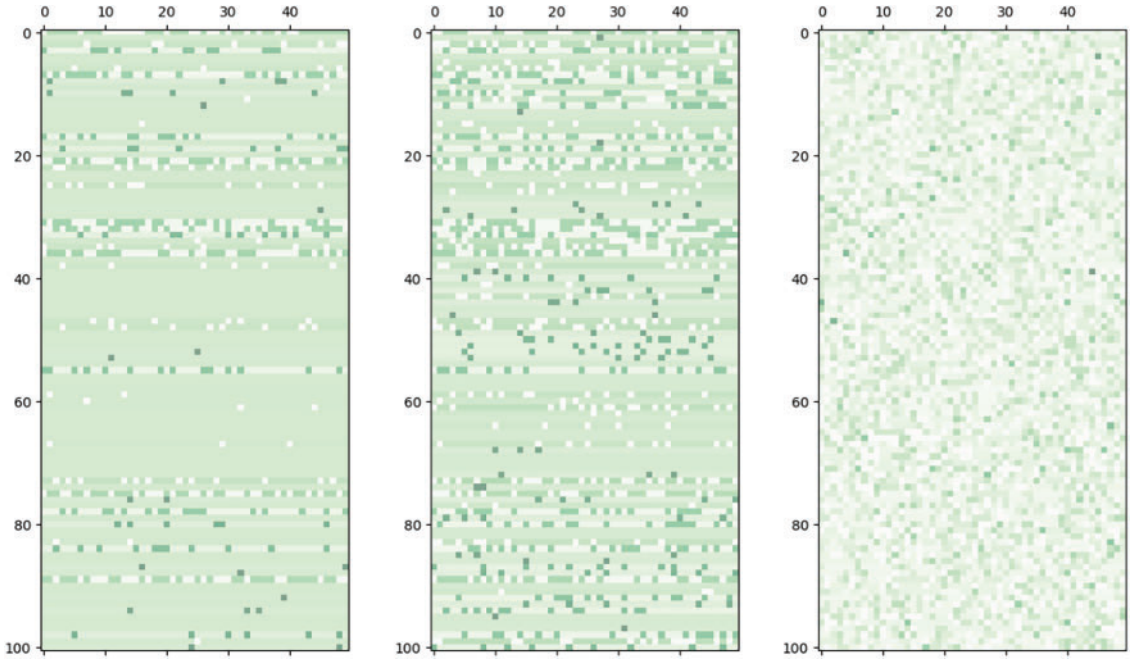

Figure 4: Dirichlet grouping of the UCF-101 dataset under different values of α

4 Experiments

In this section, experiments were conducted to verify the effectiveness of video action recognition models based on spatiotemporal features and personalized federated learning methods. The performance of the models was compared across multiple indicators and the recognition accuracy on the UCF-101 dataset was provided.

4.1 Experiments Environment

The experiments in this paper were based on the Ubuntu 22.04.1 LTS operating system, with a CPU model of Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz. GPU was used for model training and testing, with the graphics card model being NVIDIA GeForce RTX 3090 Ti, and the graphics card memory being 24 G. The experimental environment was Python 3.7.15, PyTorch 1.10, and CUDA 11.3. If the hardware conditions, especially the cache size, are reduced, it will increase the model training time.

The proposed model has a computational complexity of approximately 4.1G FLOPs and a total parameter size of about 25.6 M. With an input image size of 224 × 224, the memory usage is around 100–200 MB. The real-time performance is influenced by the hardware conditions. On the 3090 Ti GPU, the inference time of this model is approximately 5–7 ms per frame.

4.2 Datasets

The primary research objective of this paper is to investigate how to protect user data privacy in video action recognition scenarios under a personalized federated learning framework, rather than specifically optimizing the accuracy of video action recognition models. Therefore, the widely-used UCF101 dataset was adopted as the experimental dataset for this study. The UCF-101 dataset contains 13,320 videos across 101 action categories, covering a wide range of human actions in various environments. This dataset is considered one of the most comprehensive and diverse datasets for video action recognition. In the experiments of this section, the dataset were divided into training and testing sets using the holdout method, with a split ratio of approximately 7:3. To ensure the accuracy of the experimental results, the dataset was randomly divided three times and the final experimental results are the average results of the three partitioning methods. In a realistic federated learning training environment, one node corresponds to one device. The training strategy adopted in this paper is to simulate the entire federated learning process using a single device to mimic multiple nodes.

When testing the personalized federated learning method, the Dirichlet distribution method mentioned above was used to partition the non-independent and identically distributed subdatasets, simulating the federated learning scenario. This section of the experiment also tested different values of the parameter α.

4.3 Experimental Results and Analysis

In this section of the experiment, we first conducted experimental validation of the proposed spatiotemporal feature-based video action recognition method on the publicly available UCF-101 dataset. Subsequently, we verified the effectiveness of the proposed personalized federated learning-based video action recognition method on a non-independent and identically distributed (non-IID) version of the UCF-101 dataset.

4.3.1 Ablation Study

This paper proposes to obtain the difference information by subtracting the previous and next 2 frames from the sampled frame, instead of subtracting each consecutive adjacent frame separately. In addition, considering that for fast actions, subtracting frames with a large time interval may introduce significant noise to the difference information, rendering it ineffective, a pooling layer is added in the channel dimension to smooth features and extract key information. Based on these three schemes for extracting differential RGB information, comparative experiments are conducted in this section to test the performance of each scheme.

As shown in Table 1, $I_t, t \in \{1, 2, 3, 4, 5\}$ in the table represents the RGB frames at time t, where I_3 is randomly sampled for spatial feature extraction, and the other 4 frames are the 2 frames before and after time t. $Diff_{i-j}$ represents the differential information between frame I_i and frame I_j. To demonstrate the effectiveness of differential RGB, the model performance without using differential information was first tested. The $Concat$ function was used to directly concatenate the 2 frames before and after the sampled frame for information extraction. Experimental results show that the spatial-temporal module using differential information achieves better experimental results, reaching 85.851% in accuracy Top1.

Table 1: Recognition effect under different differential feature extraction methods

Number	Input data	Smooth features	Acc. Top1
1	$Concat(I_1, I_2, I_3, I_4, I_5)$	–	79.434%
2	$Concat(Diff_{1-2}, Diff_{2-3}, Diff_{3-5}, Diff_{4-5})$	No	85.444%
3	$Concat(Diff_{3-1}, Diff_{3-2}, Diff_{3-4}, Diff_{3-5})$	No	82.008%
4	$Concat(Diff_{3-1}, Diff_{3-2}, Diff_{3-3}, Diff_{3-5})$	Yes	**85.851%**

Replacing the differential between adjacent frames with the differential from sampled frames reduces accuracy. Since greater time distance between RGB frames introduces more noise, this paper applies average pooling to each frame after obtaining the differential frames. The pooled features are stacked and further smoothed through a channel-wise average pooling layer, compressing differences between features at different time points. This method improved experimental results, achieving 85.851% accuracy on the UCF-101 dataset.

In the time module, to extract action rhythm information, the translation stride during differential interaction is also an important experimental parameter worthy of consideration. Different schemes have been detailed in the previous section, and in this section, experimental results and performance analysis are directly provided.

Table 2 shows the accuracy Top1 and Top5 achieved when performing local feature differential in the time module with different translation strides. From the experimental results, it can be seen that the model with a stride of 1-1-2 achieves a higher accuracy Top1, reaching 85.931%. Compared to the original scheme 1-1-1, it improves the accuracy by 0.487%. The model with a stride of 1-2-2 obtains a 0.027% improvement in accuracy Top5 compared to the original scheme, reaching 97.159%, thereby verifying the effectiveness of global stage differential features on the UCF-101 dataset. However, when the stride is set to 1-2-3, the recognition accuracy significantly decreases, indicating that differential information with a large time span is no longer effective and may even affect recognition performance.

Table 2: Recognition performance of different translation strides

Number	Strides	Acc. Top1	Acc. Top5
1	1-1-1	85.444%	97.132%
2	2-2-2	85.038%	96.943%
3	1-1-2	**85.931%**	97.051%
4	1-2-1	85.092%	96.997%
5	1-2-2	85.363%	**97.159%**
6	1-2-3	84.686%	96.510%

4.3.2 Optimal Accuracy of Video Recognition Model

Finally, based on the best model scheme and hyperparameters obtained from the experimental tests, this paper provides the optimal recognition accuracy based on the UCF-101 dataset.

Table 3 presents a comparison of the accuracy of the proposed model with other action recognition models. Among them, the TSM model was pre-trained on simpler datasets like ImageNet. Under the same simple pre-training conditions, the proposed model achieved the highest accuracy of 87%. The TDN,

HoCNet, TSM, MEACI-NET, MTNet, and CANet models were further pre-trained on the large-scale Kinetics-400 dataset. Due to the much larger number of samples in this dataset compared to UCF-101, these models can learn more complex data representations, resulting in a significant improvement in final accuracy. Even under the condition of pre-training on both ImageNet and Kinetics-400, the proposed model still achieved the highest recognition accuracy of 97.6%.

Table 3: Comparison of the performance of the model proposed in this paper with other models

Model	Pre-training	Backbone	Acc. Top1
TSM [18]	ImageNet	ResNet50	83.2%
MANet [19]	ImageNet	ResNet50	86.2%
TDN [20]	ImageNet+Kinetics	ResNet50	97.4%
HoCNet [21]	ImageNet+Kinetics	ResNet50	94.0%
TSM [18]	ImageNet+Kinetics	ResNet50	94.5%
F2D-SIFPNet [22]	ImageNet+Kinetics	ResNet50	96.3%
MEACI-NET [23]	ImageNet+Kinetics	ResNet50	96.4%
MTNet [24]	ImageNet+Kinetics	ResNet50	96.5%
CANet [25]	ImageNet+Kinetics	ResNet50	96.6%
Our model	ImageNet	ResNet50	**87.0%**
Our model	ImageNet+Kinetics	ResNet50	**97.6%**

Specifically, the recognition accuracy for continuous actions with a strong rhythmic pattern was improved. Examples include BlowDryHair, CleanAndJerk, HorseRiding, JugglingBalls, and Rowing, shown in Fig. 5. These five actions involve the subject performing highly repetitive movements throughout the video, maintaining a certain frequency, and exhibiting a distinct action rhythm. This demonstrates the effectiveness of extracting action rhythm features through feature differences at different scales.

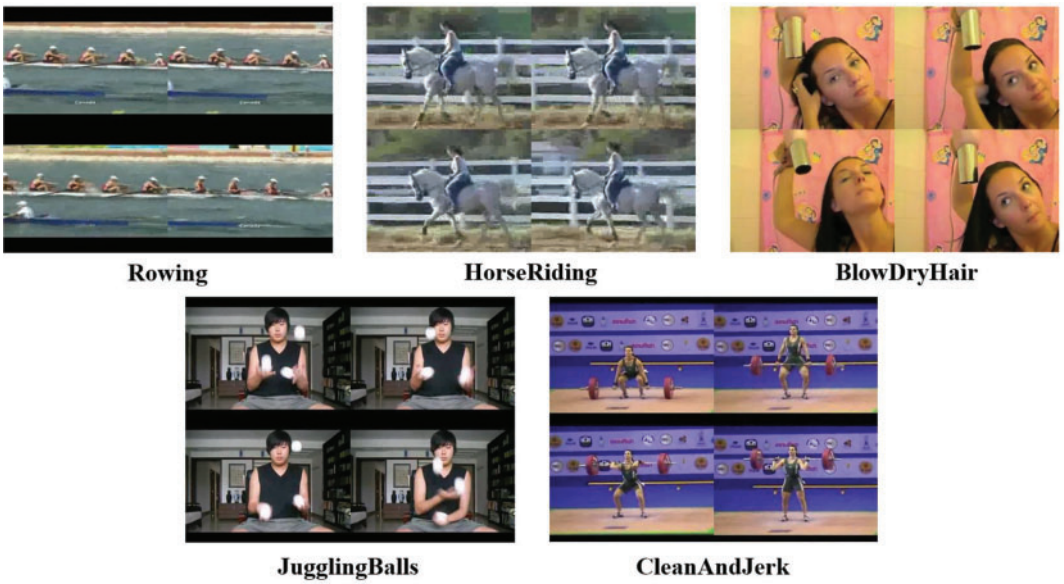

Figure 5: Diagram of significant categories of action rhythm information

In summary, the video action recognition model based on spatiotemporal features proposed in this paper can effectively improve the recognition accuracy and achieve better performance on the UCF-101 dataset. We initialized our model using the pre-trained model weights from ImageNet and Kinetics-400. ImageNet is a large-scale image dataset that contains over 14 million images, spanning 1000 categories; whereas Kinetics-400 is a large-scale video dataset that includes 400 action categories. By leveraging these pre-trained models, we ensured that our model has learned a rich and diverse feature representation, allowing it to benefit from a broader and more varied training data. This approach also enabled the accuracy of the model proposed in this paper reached the top level in the field.

4.3.3 Personalized Federated Learning Method

In this section, we first conducted tests on multiple hyperparameters of federated learning, including the number of training rounds, the number of user samples, and the degree of dataset distribution. Based on these tests, we then validated the effectiveness of the personalized federated learning mechanism proposed in this paper for video action recognition.

After deploying the dataset and model into the FedML framework, experimental tests are first conducted on the setting of hyperparameters. In this section, the total number of users C is set to 20 as a fixed parameter and kept constant. Experimental results are tested with different numbers of users sampled per round S and the number of training epochs E for each user.

As shown in Table 4, when the user training epoch is 1, the model converges after 450 communication rounds, while when the user training epoch is 5, it converges after 145 communication rounds. Although the number of communication rounds decreases, the total number of training epochs increases from 1×450 to $5 \times 145 = 725$ epochs, significantly increasing the training cost. Furthermore, when the training epoch further increases to 10, the recognition accuracy actually decreases.

Table 4: The experimental results based on different training epochs for each user

Number	Users/round	Epoch	Communication round	Acc.
1	4	1	450	96.077%
2	4	5	145	96.531%
3	4	10	85	95.696%

From the results, it can be observed that the training effect of the model under federated learning is not stable and does not steadily increase with the increase in local training epochs of users. This is due to the non-independent and identically distributed nature of the data, resulting in significant differences between the data distribution of each user and the overall dataset. In traditional deep learning training, each training epoch allows the model to learn complete data features, and the model is optimized with increasing training epochs. However, under federated learning conditions, an increase in training epochs can lead to the model learning too many individual characteristics of user local data, causing the model to overfit. This not only increases the training cost but also fails to achieve better performance. Increasing the aggregation frequency of the model can make the global model closer to the original optimal parameters.

After determining the user training epoch, this paper also conducted experiments based on different numbers of user samples per round. The test results are shown in Table 5. It can be observed that as the number of user samples increases, the training effect of the model also improves. This is because when the total number of users is fixed, the more users sampled per round, the more data participates in the training,

and the impact of each user's individual characteristics on the aggregation is reduced. The parameters aggregated by the central server tend to be more balanced. However, the training time cost will inevitably increase with the increase in the number of samples. From the perspective of simulating a real federated learning environment, the sampling value cannot be set too high. Therefore, in the following experiments, the number of sampled users per round was set to $S = 4$, meaning that 1/5 of the users (data) participate in the training each round.

Table 5: The experimental results based on different numbers of user samples per round

Number	Users/round	Epoch	Communication round	Acc.
1	2	1	480	95.448%
2	4	1	450	96.077%
3	5	1	450	96.558%

The following experiments are conducted based on different data distribution scenarios, with reference to federated learning datasets from other fields to set parameters, using two grouping methods: Dirichlet data distribution and uniform grouping. The experimental results are shown in Table 6, where $Dir(*)$ represents the Dirichlet distribution, and the parameter α controls the degree of data dispersion. In other studies, for datasets MINIST and CIFAR-10 with 10 categories, α is often set to 0.5. Since this paper's experiments use a video behavior recognition dataset with a large amount of data and sample sizes, it is necessary to experiment to test the best data grouping method. The uniform grouping data distribution matches the original dataset and is used to compare the impact of non-IID data grouping on training effectiveness.

Table 6: Experimental results based on different data distributions

Number	Data distribution	Communication round	Acc.
1	$Dir(0.5)$	480	96.023%
2	$Dir(1)$	450	96.077%
3	$Dir(10)$	420	96.377%
4	Uniform grouping	180	96.402%

From the experimental results, it can be observed that under uniform grouping, the model's convergence speed in federated learning training is the fastest. However, when using Dirichlet distribution for grouping, as the α value decreases, the data distribution becomes more scattered, requiring more rounds for model convergence. This also affects the model's optimization process, leading to suboptimal training effectiveness and impacting the highest recognition accuracy after convergence. The experimental results further illustrate the impact of unevenly distributed data storage on recognition performance in the federated learning environment. Considering the use of Dirichlet distribution for data grouping, each user's allocation of data types and quantities is completely random, resulting in fewer common features among user local data. This is suitable for scenarios where public models are used for parameter training. In practical applications, however, each user's local data often exhibits strong personalized characteristics. Similar to datasets like MINIST and CIFAR-10 with special labels, they are more conducive to personalized federated learning research. Therefore, in testing the personalized federated learning scheme, we ensure each user's training and test sets have the same sample distribution, with the same data categories proportionally represented in both sets. Based on the experimental results of hyperparameters and dataset grouping methods, the total number of users C is

set to 20, the number of users sampled in each federated learning communication round S is set to 4, and the number of local training rounds E for each user is set to 1. The dataset grouping method is $Dir(1)$. Under the above parameter settings, experiments were conducted to verify the personalized federated learning-based optimization model for video action recognition proposed in this paper, comparing the experimental results under conventional federated learning training and personalized federated learning conditions.

Table 7 presents the highest accuracy rates Top1 and Top5 achieved by the model in this paper on local datasets of 20 users under conventional federated learning and personalized federated learning. From the average accuracy rates, the personalized federated learning approach proposed in this paper achieves better results on both indicators, with the Top1 accuracy reaching 97.792%, an improvement of 1.155%, and the Top5 accuracy reaching 99.861%, an improvement of 0.079%.

Table 7: Comparison of experimental results between conventional federated learning and personalized federated learning

ID	Acc. Top1		Acc. Top5	
	FL	PFL	FL	PFL
1	96.795%	97.312%	100%	100%
2	97.701%	98.077%	100%	100%
3	97.312%	98.333%	100%	99.444%
4	97.222%	97.778%	100%	99.444%
5	95.312%	95.556%	99.479%	99.444%
6	97.396%	96.774%	99.479%	99.462%
7	95.402%	99.405%	98.851%	100%
8	97.849%	97.849%	100%	100%
9	94.444%	97.312%	100%	100%
10	94.624%	97.312%	98.925%	100%
11	94.444%	97.222%	99.444%	100%
12	94.048%	96.237%	100%	100%
13	100%	97.222%	100%	99.444%
14	96.354%	98.387%	100%	100%
15	96.774%	98.718%	100%	100%
16	99.444%	98.889%	100%	100%
17	96.774%	97.222%	99.462%	100%
18	98.925%	98.925%	100%	100%
19	96.237%	97.849%	100%	100%
20	95.699%	99.462%	100%	100%
Avg.	96.637%	**97.792%**	99.782%	**99.861%**

The experimental results validate the necessity of users holding private parameters, especially in the application scenario of surveillance video action recognition. Local user data inherently possesses strong individual characteristics. For example, in a home surveillance environment, static features such as background and main subjects can vary significantly between users and do not need to be included in the public aggregation on the central server. The proposed video action recognition method based on personalized federated learning and spatiotemporal features designates the first two layers of the network, which focus on

extracting static features, as private layers. The parameters from the subsequent three stages, which extract action features, are used for public aggregation, thereby enhancing the model's training performance.

5 Conclusion

This paper addresses the need for data privacy protection and data security in video surveillance by proposing a video action recognition method based on personalized federated learning and spatiotemporal features. First, the complete process of the personalized federated learning method and the production method of non-independent and same-distribution video action recognition datasets are introduced. Then, for video action recognition, a new spatiotemporal feature-based video action recognition algorithm is proposed, which includes two main modules: a spatial-temporal module based on local information and a time module based on global information. The local module extracts local spatiotemporal features based on each video segment while the global module interacts with local features through a differential approach on different action rhythms based on local information, and further uses neural networks to extract bidirectional action features. Subsequently, a personalized federated learning training scheme is provided. In the experimental analysis phase, multiple optional parameters for the modules were evaluated and experiments were conducted for different learning rate settings. Finally, leveraging the personalized federated learning framework, which incorporates stage-by-stage extraction of local spatiotemporal and global temporal features, the proposed method achieved an average accuracy of 97.792% on the non-independent and identically distributed UCF-101 public dataset. Additionally, a comprehensive comparison was made between the results of traditional and personalized federated learning. By processing local and global features separately without uploading users' original video data or personalized features to the central server, the risk of user privacy data leakage is minimized, making federated learning an effective mechanism for enhancing model performance while protecting user privacy.

Future work will focus on optimizing the proposed model, particularly in terms of its adaptability to various real-world scenarios. The current effectiveness of the method relies on the quality and quantity of local data, and potential improvements include introducing argumentation-based methods to enhance model interpretability. In scenarios with long-tail data distribution, some users may have limited or low-quality local data, which can constrain the training effectiveness during the local feature extraction phase and impact overall performance. Future research aims to investigate asynchronous federated mechanisms and dynamic feature calibration methods to address these issues, achieving a better balance between privacy protection and model performance.

Acknowledgement: None.

Funding Statement: This work was supported by National Natural Science Foundation of China (Grant No. 62071098); Sichuan Science and Technology Program (Grants 2022YFG0319, 2023YFG0301 and 2023YFG0018).

Author Contributions: Study conception and design: Rongsen Wu and Yuhang Zhang; data collection: Zelei Wu, Shiyang Tang and Yunji Li; analysis and interpretation of results: Jie Xu, Changming Zhao and Yiweng Xie; draft manuscript preparation: Rongsen Wu, Jie Xu, Yuhang Zhang, Changming Zhao and Jinhong Guo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data in this paper can be found in Google Scholar at https://scholar.google.com.

Ethics Approval: Not applicable.

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

References

1. Li Y, Liang Q, Gan B, Cui X. Action recognition and detection based on deep learning: a comprehensive summary. Comput Mater Contin. 2023;77(1):1–23. doi:10.32604/cmc.2023.042494.
2. Liu S, Luo Z, Li Y, Wang Y, Fu W, Ding W. Solution of wide and micro background bias in contrastive action representation learning. Eng Appl Artif Intell. 2024;133(11):108244. doi:10.1016/j.engappai.2024.108244.
3. Khean V, Kim C, Ryu S, Khan A, Hong MK, Kim EY, et al. Human interaction recognition in surveillance videos using hybrid deep learning and machine learning models. Comput Mater Contin. 2024;81(1):773–87. doi:10.32604/cmc.2024.056767.
4. Xu J, Song R, Wei H, Guo J, Zhou Y, Huang X. A fast human action recognition network based on spatio-temporal features. Neurocomputing. 2021;441(2):350–8. doi:10.1016/j.neucom.2020.04.150.
5. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag. 2020;37(3):50–60. doi:10.1109/MSP.2020.2975749.
6. Tyagi S, Rajput IS, Pandey R. Federated learning: applications, security hazards and defense measures. In: 2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT); 2023; Dehradun, India: IEEE. p. 477–82.
7. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. FoundTrends® Mach Learn. 2021;14(1–2):1–210. doi:10.1561/2200000083.
8. Aggarwal M, Khullar V, Rani S, Prola TA, Bhattacharjee SB, Shawon SM, et al. Federated learning on internet of things: extensive and systematic review. Comput Mater Contin. 2024;79(2):1795–834. doi:10.32604/cmc.2024.049846.
9. Caroprese L, Ruga T, Vocaturo E, Zumpano E. Lung cancer detection via federated learning. In: 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2023; Istanbul, Turkiye. p. 3862–7.
10. Zhao Y, Liu H, Li H, Barnaghi P, Haddadi H. Semi-supervised federated learning for activity recognition. arXiv:2011.00851. 2020.
11. Shome D, Kar T. FedAffect: few-shot federated learning for facial expression recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 4168–75.
12. Rehman YAU, Gao Y, Shen J, deGusmao PPB, Lane N. Federated self-supervised learning for video understanding. In: European Conference on Computer Vision; 2022; Tel Aviv, Israel: Springer. p. 506–22.
13. Doshi K, Yilmaz Y. Federated learning-based driver activity recognition for edge devices. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022; New Orleans, LA, USA. p. 3338–46.
14. Tu NA, Abu A, Aikyn N, Makhanov N, Lee MH, Le-Huy K, et al. FedFSLAR: a federated learning framework for few-shot action recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2024; Waikoloa, HI, USA. p. 270–9.
15. Luo Z, Fu W, Liu S, Anwar S, Saqib M, Bakshi S, et al. Cefdet: cognitive effectiveness network based on fuzzy inference for action detection. In: Proceedings of the 32nd ACM International Conference on Multimedia. MM '24; 2024; New York, NY, USA: Association for Computing Machinery; p. 7985–94.
16. Yurochkin M, Agarwal M, Ghosh S, Greenewald K, Hoang N, Khazaeni Y. Bayesian nonparametric federated QÀ learning of neural networks. In: International Conference on Machine Learning; 2019; Long Beach, CA, USA: PMLR. p. 7252–61.
17. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated learning with non-iid data. arXiv:1806.00582. 2018.
18. Lin J, Gan C, Han S. TSM: temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019; Seoul, Republic of Korea. p. 7083–93.
19. Li X, Yang W, Wang K, Wang T, Zhang C. Manet: motion-aware network for video action recognition. Comp Intell Syst. 2025;11(3):167. doi:10.1007/s40747-024-01774-9.
20. Wang L, Tong Z, Ji B, Wu G. TDN: temporal difference networks for efficient action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 1895–904.
21. Dong W, Wang Z, Zhang B, Zhang J, Zhang Q. High-order correlation network for video recognition. In: 2022 International Joint Conference on Neural Networks (IJCNN); 2022; Seoul, Republic of Korea: IEEE. p. 1–7.

22. Ming Y, Zhou J, Jia X, Zheng Q, Xiong L, Feng F, et al. F2D-SIFPNet: a frequency 2D Slow-I-Fast-P network for faster compressed video action recognition. Appl Intell. 2024;54(7):5197–215. doi:10.1007/s10489-024-05408-y.
23. Li B, Chen J, Zhang D, Bao X, Huang D. Representation learning for compressed video action recognition via attentive cross-modal interaction with motion enhancement. arXiv:2205.03569. 2022.
24. Sheng X, Li K, Shen Z, Xiao G. A progressive difference method for capturing visual tempos on action recognition. IEEE Transact Circ Syst Video Technol. 2022;33(3):977–87. doi:10.1109/TCSVT.2022.3207518.
25. Gao X, Chang Z, Ran X, Lu Y. CANet: comprehensive attention network for video-based action recognition. Knowl Based Syst. 2024;296(8):111852. doi:10.1016/j.knosys.2024.111852.