INFORMATICA, 2025, Vol. 36, No. 4, 985–1012

DOI: https://doi.org/10.15388/25-INFOR613

Key Frame-Based Skeleton Extraction for Lightweight Human Action Recognition Networks

Leiyue YAO[1], Chao ZENG[1], Jianying XIONG[1,*], Keyun XIONG[1],
Lei ZHANG[2], Yucheng WANG[3]

[1] *College of Computer Science, Jiangxi University of Chinese Medicine, Nanchang, China*
[2] *Hanlin Hangyu (Tianjin) Industrial Co., Ltd., Tian Jin, China*
[3] *The University of Melbourne, Melbourne, VIC, Australia*
e-mail: special8212@sohu.com

Received: May 2025; accepted: November 2025

Abstract. Human Action Recognition (HAR) is an important task in computer vision with diverse applications. However, most existing methods rely on all frames of an action video for classification, which leads to high computational cost and low efficiency. In many cases, a compact set of key keyframes can effectively encode the essence of a complete action. Therefore, this study proposes an efficient HAR method that combines a new keyframe extraction algorithm with a lightweight neural network. Our contribution is three-fold. Firstly, an accurate and efficient key frame algorithm is proposed to alleviate the issue of frame-order confusion in classical clustering methods. Secondly, a key-frame-based multi-feature fusion matrix is constructed to address information loss from spatio-temporal trajectory overlap and the sensitivity issue of viewpoint changes in classical models. Thirdly, a lightweight neural network model is designed to achieve effective convergence within a short training period. The proposed method was evaluated on two public datasets (UTKinect-Action3D and Florence-3D) and a self-collected dataset (HanYue-3D). The experiment results show the advantages of our method in both accuracy and efficiency.

Key words: key frame extraction, lightweight neural network, multi-scale learning, skeleton-based action recognition, CNN-based action recognition.

1. Introduction

HAR has always been regarded as a core task in computer vision and pattern recognition. Its applications include intelligent monitoring (Chen *et al.*, 2023), security protection (Biswal *et al.*, 2024), sports analysis (Zhao *et al.*, 2025) and human-computer interaction (Yu *et al.*, 2024). These fields often require efficient and accurate video understanding. However, human action videos are typically long and complex, and often contain redundant information for HAR. To address this issue, many researchers have begun to study how to extract key frames with rich information (Lanzoni *et al.*, 2024; Liu *et al.*, 2024).

In recent years, Convolutional Neural Networks (CNNs) have benefited from their strong ability in image feature extraction and pattern recognition to provide solid and

*Corresponding author.

extensive technical support for HAR (Kong and Fu, 2022). At the same time, Simonyan and Zisserman (Simonyan and Zisserman, 2014) found that effective keyframe extraction and preprocessing can significantly improve the performance of action recognition. Researchers have been trying to find effective and unique frame extraction methods and representations and improve them to increase the accuracy of HAR. Early HAR methods relied on Motion History Images (MHI), Static History Images (SHI), and Motion Energy Images (MEI). However, SHI performs poorly with occlusions and deformations. MHIs and MEIs have problems such as loss of time information and cumulative error. Consequently, these methods have gradually fallen out of favour in the field of HAR. With the increasing popularity of the Kinect camera and the development of deep learning algorithms such as OpenPose, the acquisition of human skeleton data has become both efficient and accurate. Generally, skeleton data can be represented in either two dimensions (2D) or three dimensions (3D). However, compared with 3D skeleton data, 2D skeleton data lacks depth information, which may limit its ability to capture complex human motions. The accuracy of action recognition based on 3D skeleton features is usually higher than that based on 2D skeleton features. Therefore, 3D skeleton features are widely used in HAR (Khezerlou *et al.*, 2023).

Researchers have developed a variety of effective representations for skeleton data analysis and modelling. The well-known methods include Skeleton Motion Mapping (SMM) and Joint Trajectory Mapping (JTM). Through these new motion representations and the powerful spatial feature extraction capabilities of CNNs, the accuracy of HAR has gradually improved. Compared with recurrent architectures such as LSTM and BiLSTM, CNNs pay more attention to local spatial correlations and perform more efficiently in feature extraction (Dutta *et al.*, 2025). The recurrent architectures usually perform better in temporal modelling of input data. Inspired by the characteristics of human visual attention, researchers have proposed the concept of attention mechanism that enables models to selectively emphasize information-rich features while suppressing irrelevant features. Among them, lightweight modules such as the Convolutional Block Attention Module (CBAM) can achieve fine processing in channels and spatial domains. Transformer-based architectures are good at modelling global dependencies and have demonstrated remarkable performance across various sequence learning tasks (Xin *et al.*, 2023). However, long-standing challenges such as spatio-temporal overlap and viewpoint sensitivity at the input stage remain difficult to address satisfactorily. The recognition accuracy largely depends on low-noise data.

Recognizing the same behaviour will significantly increase the recognition difficulty under different perspectives, body proportions and lighting conditions (Zhu *et al.*, 2023). In addition, the recognition task becomes even more challenging due to the differences in the movement styles of individuals. Meanwhile, frame-by-frame video analysis methods face three major difficulties in motion feature extraction: excessive computation, serious data redundancy, and inefficient processing speed (Zhao *et al.*, 2024), especially when distinguishing subtle differences between similar behaviours such as walking and running, walking forward and backward, and sitting and standing (Alsaadi *et al.*, 2025). These behaviours require the proposed method be able to accurately capture the subtle differences in motion patterns. The use of appropriate key frames in HAR tasks can further improve the recognition accuracy. However, researchers face two basic challenges in keyframe extraction: accurate selection of key frames and extraction of a unified number of key frames. At present, keyframe extraction methods are mainly divided into three categories: 1) shot-boundary detection methods, 2) sampling and clustering techniques, and 3) deep learning-based frame selection (Kar *et al.*, 2024; Gan *et al.*, 2023; Zhang, 2024). Although these methods improve processing efficiency, they have serious limitations: clustering-based methods will lead to temporal disorder, and deep learning based methods generally lack sufficient interpretability. The core challenge of keyframe extraction is to optimize computational efficiency without reducing recognition accuracy.

In this paper, we propose a key frame clustering algorithm (KKF), which solves the problems of overlooking temporal sequences and limited accuracy in unsupervised keyframe clustering. We further extract relative motion features to construct a key dense joint motion matrix (KDJMM) based on the selected keyframes. The new feature construction method significantly improves the recognition efficiency while maintaining the recognition accuracy. In addition, a lightweight convolutional neural network (L-KFNet) is constructed to learn from KDJMM. Compared to classical CNNs, L-KFNet achieves faster convergence and higher accuracy.

The structure of this paper is that Section 2 introduces the latest research progress of HAR based on CNN and keyframe extraction; Section 3 introduces the KKF algorithm design, KDJMM construction, and L-KFNet model in detail; Section 4 presents and discusses the experimental results; Section 5 provides an overview of future research works.

2. Related Work

The existing methods have achieved notable results in extracting spatiotemporal features, but there are still some issues. Firstly, many methods rely on fixed image construction or basic key-frame extraction strategies. These methods are difficult to maintain both a consistent number of key frames and temporal continuity, which leads to suboptimal performance in handling dynamic, continuous, and similar actions. Secondly, classical clustering and feature extraction methods have limited adaptability with respect to temporal consistency and action diversity while capturing certain spatial variations. Therefore, they are insufficient for similar action recognition and few-shot learning tasks (FSL).

2.1. *CNN-Based HAR Methods*

Du *et al.* (2015) addressed the problem of trajectory overlapping. They constructed colour motion pseudo-images of uniform size by using a skeleton sequence (x, y, z) mapped sequentially to the three channels of a colour image (R, G, B). The motion information is placed sequentially in non-overlapping slices. Later, Yang *et al.* (2018) mentioned that human skeletal joints are not isolated from each other. Therefore, they proposed a tree-shaped skeleton joint arrangement, which better preserves spatial information. Liu *et al.* (2019) proposed a novel spatio-temporal image representation of skeleton joints called

"Skepxels." The method utilizes basic building blocks called "Skepxel" to construct skeleton images of arbitrary dimensions. These images encode fine-grained spatiotemporal information about the human skeleton across different frames. Additionally, multiple joint permutations are employed to enhance action recognition using spatio-temporal visual attention on skeleton image sequences. Ke *et al.* (2017) proposed a skeletal representation. Four base points are selected, and the 3D information of their connection points is represented in three slices each. Twelve grayscale images are then generated and fed into a multi-task learning network to learn the internal relationships among the selected base points for action classification. Caetano *et al.* (2019) noted that CNNs implicitly extract temporal human motion information from skeleton images. The position information and motion direction of the skeleton joints are computed by setting the period to capture the long-term joint interaction information in the action. Meanwhile, it introduces the arrangement of tree skeleton joints (Yang *et al.*, 2018) to construct a new skeleton motion representation.

These methods partly solved the spatio-temporal overlapping problem, but some significant limitations remain: 1) Constructing a colour image requires the feature values to be within the range of 1–255, which makes it difficult to distinguish subtle actions and may obviously reduce feature accuracy. 2) Simple scaling and cropping can deform the action representation to achieve a uniform size required by CNN input. For example, a running action may be misclassified as walking when forced to resize to a fixed dimension as a colour motion image. To solve these problems, our previous work (Yao *et al.*, 2022) proposed to utilize more joint information, displacements, angles, velocities, etc., to construct a compact 3D Dense Joint Motion Matrix (DJMM) in the form of floating-point numbers. This forms a DJMM with dimensions: number of joints $\times$ number of features $\times$ frame size, ensuring the preservation of raw data. However, efficiently generating a uniform-size input remains an urgent challenge.

2.2. *Key Frame Extraction Method*

At present, key frame extraction methods for HAR are mainly evolved from traditional summary extraction techniques. Due to the rich information in images, hand-crafted features such as HOG, SSIM, SIFT, and optical flow have been developed. Researchers have conducted extensive work on key frame extraction in RGB video (Zhao *et al.*, 2024), which can be categorized into three parts: shot-based, sampling, and clustering-based techniques. Inspired by research in RGB video and the need for more efficient HAR, researchers began to study key frame extraction from human action videos. Li *et al.* (2023) identify 3D skeleton action sequences based on a series of meaningful movements, including changes in body posture direction and geometric features and improve motion analysis efficiency. Phan *et al.* (2021) constructed a motion variation curve by calculating the pixel intensity change between two consecutive frames and selecting the k highest localized extremes as the corresponding key frames.

The inter-frame sampling strategy is a commonly used method to extract key frames. However, it will likely lose frames with rich information. Similarly, simply using the information difference between adjacent frames to select key frames relies only on local information and fails to capture long-term dependencies. Thus, more and more researchers are exploring and proposing new algorithms to identify the most suitable key frames. For example, Gan *et al.* (2023) utilized deep learning techniques to select key frames. They extracted the motion and appearance information of each frame through CNN, labelled each frame with a score and fitted a curve. Then, the local maxima and minima on the curve were identified and corresponded to the key frames in the video. Zhao *et al.* (2023) proposed an improved k-means algorithm based on joint contribution weighting, which makes the key frame extraction pay more attention to the joints with more significant relative motion. For the skeleton data, k frames with the maximum inter-frame euclidean distance and their corresponding previous frames are selected. They are used to construct the left and right boundaries to form k clusters for initialization. One frame is selected as the cluster centre, and the euclidean distance is used for iteration until the centre no longer changes. Zhang *et al.* (2023) proposed a video key frame extraction method based on multi-feature fusion of quaternion fourier transform. The proposed method first extracts dynamic and static features from colour video sequences. Their fused phase spectra are then obtained according to the quaternion fourier transform. Gaussian filtering is used to remove noise in the fused phase spectra. Then, the inverse fourier transform is performed to obtain the fused feature map. Finally, an adaptive key frame selection criterion is constructed. The final key frame set is obtained by feature map selection. Chen *et al.* (2024) transformed the key frame selection problem into a multi-objective optimization problem for binary coding. An evaluation model based on domain information and the number of key frames was proposed, which can adaptively adjust the number of key frames according to the compression rate while preserving the temporal order of actions.

The method of cluster-based key frame extraction is superior to basic difference-maximization approaches. The reason for its advantage is that it iterates clusters by using spatial features as indicators and extracts a consistent number of key frames between different action types. This explains why it is widely adopted in HAR systems. However, clustering methods overemphasize spatial feature variations. This can easily lead to incorrect grouping of frames that are distant in time but similar in space. This will disrupt the preservation of action phase transitions (e.g. take-off vs. landing in jumps). Researchers have attempted to introduce temporal modelling. However, these approaches either require adjusting the threshold for specific actions or cannot maintain a uniform number of key frames between different action types.

3. Our Method

Figure 1 shows the overall framework of our method. The design of key frame extraction method is introduced firstly. We calculate various local relative features based on the key frames to form the KDJMM. To handle the constructed KDJMM input, we design the L-KFNet for processing and classification. This combination of key frames and the shallow CNN significantly reduces input parameters and model training complexity.

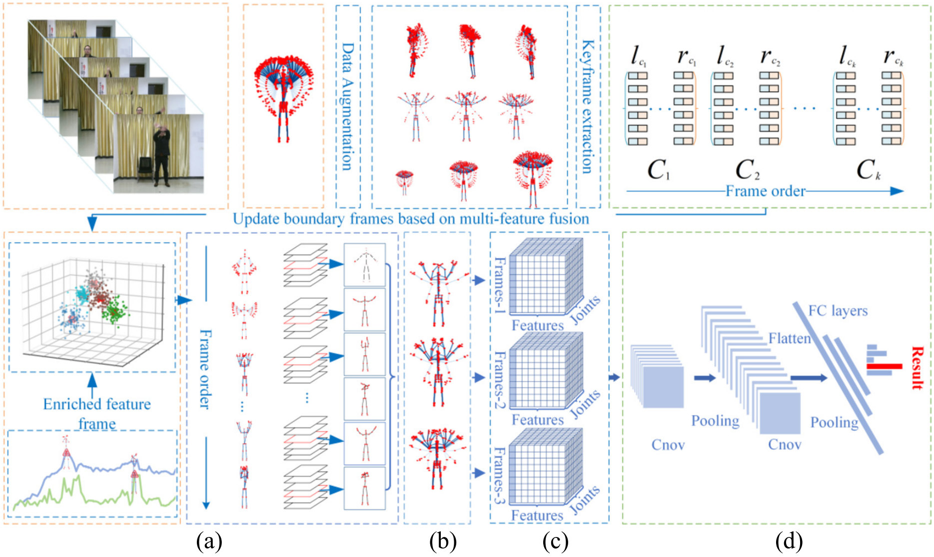


Fig. 1. Overall framework of the method. (a) Key frame selection module. (b) Multi-scale training method. (c) Feature representation module. (d) Shallow CNN for HAR.

The method improves the efficiency of HAR. Furthermore, we propose a simple and effective data augmentation method for a small-sample dataset of human behaviour. A multi-scale learning strategy is also employed during training so that the network can learn behavioural information at different time scales. The specific methods are described as follows:

3.1. *Key Frame Extraction Algorithm (KKF)*

A drawback of unsupervised key frame based clustering algorithms is that they rely solely on the euclidean distance between skeleton joints when performing clustering. This method may group frames that are far apart in time based on pose similarity, thereby disrupting the inherent temporal continuity of actions. These methods often disrupt temporal order during clustering, which can lead to misjudgments in HAR. Many current algorithms that maintain temporal stability of actions face a critical limitation, as they either fail to achieve scale-consistent key frames or introduce random errors during scale normalization. These methods often rely on random frame deletion or insertion, which carries the risk of losing informative frames and adding low-saliency frames. To address these issues and establish network inputs with a unified scale without compromising input quality, we propose the KKF algorithm.

The proposed KKF algorithm aims to select the K key frames that can best represent the motion trend of the entire action. Inspired by the previous experiments (Zhang *et al.*, 2023), we employed the concepts of inter-frame difference and cluster mean frame. As indicated in Fig. 2, we take 15 skeleton joints as an example, and the position data of

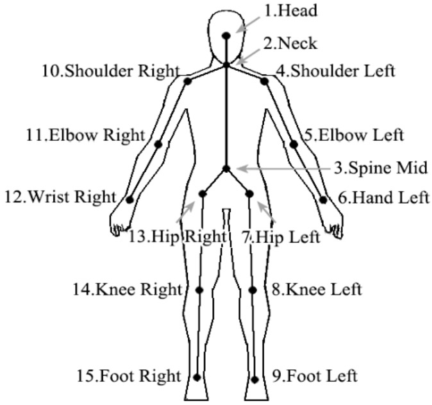


Fig. 2. Fifteen joints of the human skeleton, used as original points for feature extraction.

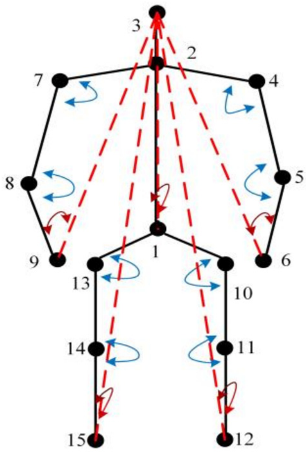


Fig. 3. Joint angles are used for our proposed method. Each joint is used to construct our KDJMM.

these joints in each frame are represented as the following 45-dimensional feature vector. In all the following equations, the i represents the frame number of the human behavioural skeleton, the n represents the n-th skeletal joint point of each frame, and the position information of each frame F_i can be expressed as:

$$F_i = \{(x_i^1, y_i^1, z_i^1), \dots, (x_i^n, y_i^n, z_i^n), \dots, (x_i^{15}, y_i^{15}, z_i^{15})\}. \tag{1}$$

Although inter-frame difference methods effectively capture motion magnitude, they exhibit limitations in detecting subtle kinematic patterns like hand clapping. This paper introduces a hybrid method that combines joint angle and joint position features. It improves keyframe extraction accuracy, especially in fine-grained action recognition tasks. As shown in Fig. 3, we select the angle of each joint and map it across the three planes using the calculation methods in formulas (16) and (17). The angle information of each frame can also be represented A_i as a 45-dimensional vector:

$$A_i = \{(\theta_{xi}^1, \theta_{yi}^1, \theta_{zi}^1), \ldots, (\theta_{xi}^n, \theta_{yi}^n, \theta_{zi}^n), \ldots, (\theta_{xi}^{15}, \theta_{yi}^{15}, \theta_{zi}^{15})\}. \tag{2}$$

When extracting key frames, we apply min-max normalization to standardize the scale of both angle and position features to a [0, 1] range. This process ensures compatibility between features that originally have divergent value ranges (e.g., joint angles in radians vs. displacement in meters), allowing their joint utilization in the unified key frame selection metric.

$$W(X, Y) = \frac{X - MIN(Y)}{MAX(Y) - MIN(Y)} Y \in \{F_i, A_i\}, \quad X \in \{Y\}, \tag{3}$$

$$W_i = \{W^1(x_i^1, F_i), W^2(y_i^1, F_i), W^3(z_i^1, F_i), \ldots, W^{45}(z_i^{15}, F_i), W^{46}(\theta_{xi}^1, A_i), \ldots, W^{90}(\theta_{zi}^{15}, A_i)\}. \tag{4}$$

The normalized data of each frame are summed to form a 90-dimensional vector W_i for calculating the key frame algorithm index.

The inter-frame difference ($V_{i,j}$) and frame-to-CMF distance (V_{i,c_j}) can be expressed as

$$V_{i,j} = \sum_{n=1}^{90} \sqrt{(W_j^n - W_i^n)^2}, \tag{5}$$

$$V_{i,c_j} = \sum_{n=1}^{90} \sqrt{(CMF_{c_j}^n - W_i^n)^2}, \tag{6}$$

where the W_i^n represents the n-th dimension of the normalized vector and $CMF_{c_j}^n$ represents the n-th dimension of the CMF_{c_j}.

This indicates that a larger V value corresponds to a greater comprehensive difference between frames. The core objective of our KKF algorithm is to cluster frames with high spatio-temporal similarity into one category, while assigning frames with significant spatio-temporal differences to distinct clusters. To ensure temporal stability, the KKF algorithm first divides all frames evenly into K initialization intervals based on temporal order to form K clusters $C_i = \{c_1, c_2, \ldots, c_{k-1}, c_k\}$. The first and last frames of each initialization interval are selected to form two sets: The right boundary set of the initialization cluster $\{r_{c_1}, r_{c_2}, \ldots, r_{c_{k-1}}, r_{c_k}\}$, and the left boundary set $\{l_{c_1}, l_{c_2}, \ldots, l_{c_{k-1}}, l_{c_k}\}$. This method satisfies the requirement of significant spatial differences in cluster initialization to reduce the time consumption of iterations.

The concept of a cluster mean frame (CMF) is proposed and defined in equations (7) and (8):

$$N_{c_i}^{n} = \sum_{t=l_{c_i}}^{r_{c_i}} \frac{W_t^n}{(r_{c_i} - l_{c_i})}, \tag{7}$$

$$CMF_{c_i} = \{N_{c_i}^1, N_{c_i}^2, N_{c_i}^3, \ldots, N_{c_i}^{88}, N_{c_i}^{89}, N_{c_i}^{90}\}, \tag{8}$$

where the l_{c_i} and r_{c_i} are the left and right boundaries of cluster c_i, the $N_{c_i}^n$ is the n-dimensional vector of the CMF of cluster c_i. The CMF is derived from each dimension and mean of all frames in cluster c_i, which is the vector that can best comprehensively represent the spatial information in the cluster frame. The CMF facilitates computation of the inter-cluster and frame-to-cluster dissimilarities so that the obtained frames can optimally characterize the cluster, ensuring that the holistic action semantics is obtained to the greatest extent. To operationalize this principle, we iteratively refine the attribution of the boundary frames of adjacent clusters by calculating the CMF of each cluster to achieve the balance of spatial information for each cluster and avoid cross-time compression. Finally, we iteratively optimize the cluster boundary frames and select the frame closest to the cluster CMF as the key information frame. In this way, we can extract the key frames that represent human behaviour. The pseudo-code of the critical moment mode selection algorithm is given in Algorithm 1 (i.e. KKF).

The proposed method optimizes the trade-off between accuracy and efficiency by dynamically weighting kinematic metrics. During human action analysis tasks, there exist some classic key frames predefined by human kinematics. These moments are manifested as: 1) peak kinematic discontinuity, and 2) maximum spatial deviation. These frames are rich in dynamic information and crucial for understanding the whole action sequence. For example, we can define the peak kinematic discontinuity of human movement as: the frame with the most extensive variation $S(x)$ from the previous frame, while the maximum spatial deviation as the frame with the most extensive variation $L(x)$ from the first frame. We introduce two functions: $S(x)$ and $L(x)$. Specifically, $S(x)$ denotes the short-term motion difference between consecutive frames, formulated as

$$S(x) = V_{x-1,x}, \quad x \geqslant 2. \tag{9}$$

In contrast, $L(x)$ denotes the long-term spatial deviation of frame x relative to the initial frame, defined as

$$L(x) = V_{1,x}, \quad x \geqslant 1. \tag{10}$$

Taking the action of waving the hand twice as an example, the change curves of both movement and maximum spatial deviation are visualized in Fig. 4 $S(x)$ and $L(x)$. When a peak is detected in the curve, we directly set that cluster's CMF value to the value of the special frame.

Algorithm 1 Skeleton key frame extraction(KKF)

Input:Initialization of clusters; // uniformly initialised clusters
Output: k key frames;
begin
repeat
Calculate the *CMF* for each cluster using Eq. (8)
for $j = 1$ to $K-1$ do //processing right border frames
$a = r_{c_j}, x = CMF_{c_j}, y = MF_{c_{j+1}}$
Calculate the $V_{a,x}$, $V_{a,y}$ from r_{c_j} to CMF_{c_j}, $CMF_{c_{j+1}}$ using Eq. (6)
while $V_{a,x} > V_{a,y}$ //assign boundary frames to the cluster
do
Move r_{c_j} to C_{j+1}
Update CMF_{c_j} and $CMF_{c_{j+1}}$ using Eq. (8)
Calculate the distances $V_{a,x}$, $V_{a,y}$ from r_{c_j} to CMF_{c_j}, $CMF_{c_{j+1}}$
end
end
for $j = 2$ to K do //processing left border frames
$a = l_{c_j}, x = CMF_{c_{j-1}}, y = CMF_{c_j}$
Calculate the distances $V_{a,x}$, $V_{a,y}$ from l_{c_j} to $CMF_{c_{j-1}}$, CMF_{c_j} using Eq. (6)
while $V_{a,x} < V_{a,y}$
do
Move l_{c_j} to C_{j-1}
Update $CMF_{c_{j-1}}$, CMF_{c_j} using Eq. (8)
Calculate the distances $V_{a,x}$, $V_{a,y}$ from r_{c_j} to CMF_{c_j}, $CMF_{c_{j-1}}$
end
end
until the *CMF* of each cluster does not change
Compute *KF* in each cluster that has the smallest V with the *CMF*.
Return *KF*; //key frames array
end

3.2. *Key Dense Joint Motion Matrix (KDJMM)*

Following our previous work (Yao *et al.*, 2022), the DJMM has solved the problem of motion information loss when joint motion trajectories overlap. Inspired by this approach, we use the key frames obtained by the KKF algorithm to extract features such as displacement, angle, velocity, and angular velocity of the corresponding skeleton. The feature calculations are based on relative coordinates, which enhances viewpoint robustness. Finally, the features are arranged into a uniform-scale KDJMM, allowing us to better capture the subtle movement changes.

The construction of KDJMM is shown in Fig. 5. Each frame contains information on 15 3D skeletal joints. Among them, 19 features are extracted for each joint on position,

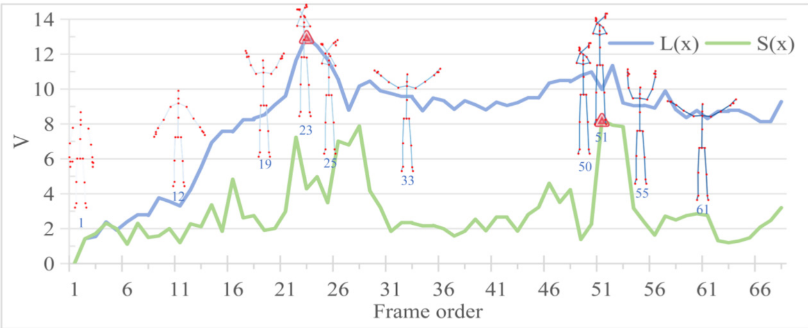
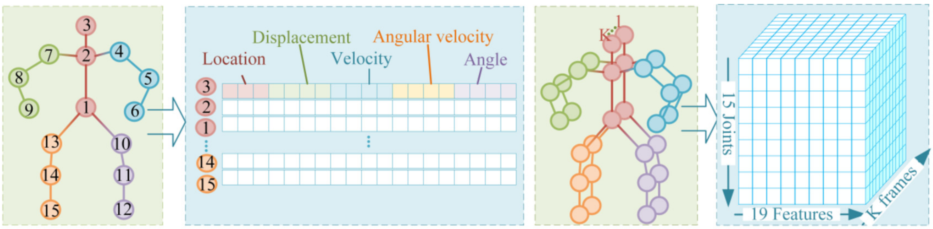


Fig. 4. 10 key frames extracted by angle and displacement with action waving as an example, $S(x)$, $L(x)$ curves, the red triangle marked is the special key frame.



(a) (b)

Fig. 5. KDJMM (a) Extracting feature information from a key frame. (b) Extracting skeletal joint information from K key frames to construct the KDJMM.

displacement, velocity, angle, and angular velocity, demonstrated in Fig. 5(a). Then, the constructed K 2D matrices are stacked together in the order of the extracted K key frames to form a temporally stable and uniform KDJMM, as shown in Fig. 5(b). The details of the feature calculation procedure are given below:

3.2.1. *Relative Coordinate Feature*

Relative coordinates are less sensitive to changes in viewing angle than absolute coordinates. Therefore, the features we extract are based on relative coordinates. Meanwhile, to enhance the fine-grained description of human action, we introduce a variety of geometric features, including inter-joint angles, velocities, displacements, and other features, as well as their mapping in the three planes.

The physical significance of the position information of each joint is to provide us with the most basic spatial information about human behaviour. The following method is used to calculate the most basic relative coordinates:

$$\begin{aligned} x_{il}^n &= x_i^n - x_1^1, \\ y_{il}^n &= y_i^n - y_1^1, \\ z_{il}^n &= z_i^n - z_1^1. \end{aligned} \tag{11}$$

In the calculation method, the x_{il}^n denotes the x relative coordinate of the n-th joint of the i-th frame we calculated, the x_i^n is the original coordinate of the n-th joint of the i-th frame, and the x_1^1 denotes the original coordinate of the first joint of the first frame. Similarly, the relative positions of all the joints are calculated. Three feature values for each joint are obtained.

3.2.2. *Displacement Feature*

The physical significance of the total displacement of the joints of each frame from the first frame is that it provides a long-distance temporal dependence to observe the human body movement from a global perspective. At the same time, in order to get a more detailed and comprehensive change situation, we map the change situation to the three planes ($x - y$, $y - z$, $z - x$). The three-plane mapping of each joint and the method of calculating the total displacement are given below:

$$\begin{aligned} Dx_i^n &= x_{il}^n - x_{1l}^n, \\ Dy_i^n &= y_{il}^n - y_{1l}^n, \\ Dz_i^n &= z_{il}^n - z_{1l}^n. \end{aligned} \tag{12}$$

The Dx_i^n represents the displacement in the X-plane of the n-th joint of the i-th frame and the corresponding joint of the first frame, and the x_{il}^n represents the relative position of the n-th joint of the i-th frame. The Dy_i^n and Dz_i^n are calculated in the same way.

The total displacement is calculated as follows:

$$D_i^n = \sqrt{\left(Dx_i^n\right)^2 + \left(Dy_i^n\right)^2 + \left(Dz_i^n\right)^2}. \tag{13}$$

The formula D_i^n denotes the total displacement of the n-th joint of the i-th frame from the joint corresponding to the first frame. The four feature values for each joint are obtained with the above formula (12)–(13).

3.2.3. *Velocity Feature*

The intensity of displacement change per unit time of each skeletal joint is expressed in a physical sense as the velocity of each skeletal joint of the human body. The eigenvalues of the joint velocity components in the x, y, and z directions are computed as follows:

$$\begin{aligned} Vx_i^n &= x_{il}^n - x_{i-1l}^n, \\ Vy_i^n &= y_{il}^n - y_{i-1l}^n, \\ Vz_i^n &= z_{il}^n - z_{i-1l}^n, \end{aligned} \tag{14}$$

where the Vx_i^n, Vy_i^n, and Vz_i^n denote the mapping values of the n-th joint in the i-th frame in the three-plane velocity. The x_{i-1l}^n, y_{i-1l}^n, and z_{i-1l}^n denote the three-dimensional relative coordinate values of the n-th joint in the i-th frame. We define the three mapping values of the feature value to be zero when i is zero.

The velocity feature is calculated as follows:

$$V_i^n = \sqrt{\left(Vx_i^n\right)^2 + \left(Vy_i^n\right)^2 + \left(Vz_i^n\right)^2}. \tag{15}$$

3.2.4. *Angle Feature*

The physical meaning of an angle is that it provides a geometrical feature of the human body structure and represents the geometrical relationship between neighbouring joints. In our definition, an angle is composed of three joints. We preferentially define two neighbouring joints as the corresponding angles of the intermediate joints. The combination of angles is identified by us as (J^2, J^1, J^3), (J^1, J^2, J^3), (J^2, J^3, J^1), (J^2, J^4, J^5), (J^4, J^5, J^6), (J^5, J^6, J^3), (J^2, J^7, J^8), (J^7, J^8, J^9), (J^8, J^9, J^3), (J^1, J^{10}, J^{11}), (J^{10}, J^{11}, J^{12}), (J^{11}, J^{12}, J^3), (J^1, J^{13}, J^{14}), (J^{13}, J^{14}, J^{15}), (J^{14}, J^{15}, J^3), where each combination represents the angle of the joint at its intermediate position. Taking (A, B, C) as an example, we calculate the angle between $\overrightarrow{\mathrm{BA}}$ vector and $\overrightarrow{\mathrm{BC}}$ vector as the angle feature of joint B as follows:

$$\begin{aligned}\overrightarrow{\mathrm{BA}} &= (A_x - B_x, A_y - B_y, A_z - B_z,),\\ \overrightarrow{\mathrm{BC}} &= (C_x - B_x, C_y - B_y, C_z - B_z).\end{aligned} \tag{16}$$

The vectors $\overrightarrow{\mathrm{BA}}$ and $\overrightarrow{\mathrm{BC}}$ are calculated by subtracting the values of the B joints using the skeletal joints A and C, respectively.

The angle is calculated as follows:

$$\theta_B = \frac{180}{\pi} \times \cos^{-1}\left(\frac{\overrightarrow{\mathrm{BA}} \cdot \overrightarrow{\mathrm{BC}}}{\|\overrightarrow{\mathrm{BA}}\| \times \|\overrightarrow{\mathrm{BC}}\|}\right). \tag{17}$$

The θ_B is the angular feature of the joint B. The dot product of vectors $\overrightarrow{\mathrm{BA}}$ and $\overrightarrow{\mathrm{BC}}$ is calculated by removing the product of the modes of the two vectors $\|\overrightarrow{\mathrm{BA}}\| \times \|\overrightarrow{\mathrm{BC}}\|$ to get the cosine value and then converted by the inverse cosine function $\cos^{-1}$ to get the radians to get our angular feature θ_B. We calculate the three-plane angular feature by choosing a two-by-two combination of coordinates of $\overrightarrow{\mathrm{BA}}$ and $\overrightarrow{\mathrm{BC}}$ vectors and then using the above angular formula to get the three-plane mapping eigenvalue of the angle. The change in angle per unit of time for each joint and the mapping on the three planes are computed to obtain the three mapping values for each joint. In the end, 19 features are calculated for each joint in proposed method. The main features are focused on displacement and angle, of which we choose 15 joints, and finally form a $K \times 15 \times 19$ size of the KDJMM. Our framework enables task-adaptive kinematic feature computation (displacement, velocity, direction, acceleration) through configurable joint selection. Specifically for gesture recognition, incorporating detailed hand joints (e.g. metacarpophalangeal) improves recognition accuracy compared to using only major limb joints.

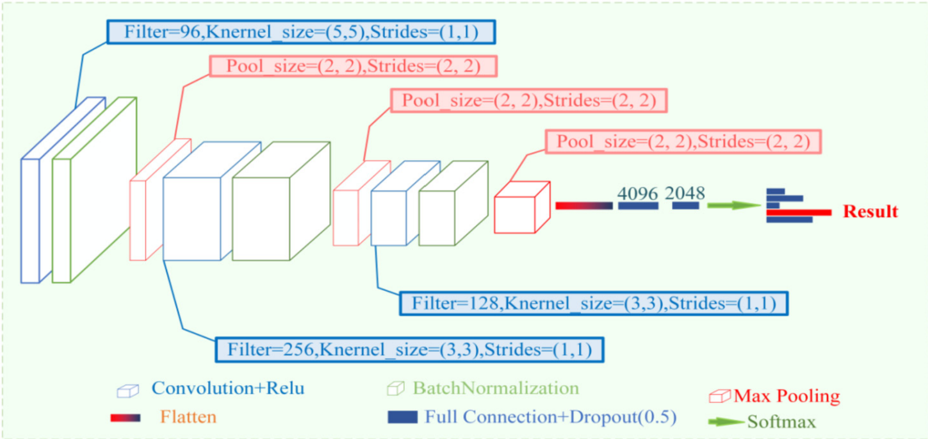


Fig. 6. Lightweight shallow convolutional neural network (L-KFNet).

3.3. *Lightweight Shallow CNN (L-KFNet)*

Deeper CNNs often demonstrate stronger learning capabilities on the constructed data, but they also require a larger input scale. This is because, during continuous convolution and pooling, the matrix scale progressively decreases until convolution becomes unfeasible. Although solutions were provided (residual blocks), deeper architectures inevitably demand more computational resources. To overcome this computational limitation, we designed a lightweight shallow neural network, retaining only three convolutional layers. Considering the size of the KDJMM, we reduced the kernel size, strides, and pooling window. Additionally, to accelerate convergence, we added a batch normalization layer after each convolutional layer. The specific network structure is shown in Fig. 6. In our experiments, we replace the final maximum pooling and flattening layers in the network with Spatial Pyramid Pooling (SPP) layers for multi-scale purpose. To further improve the representation quality, we introduced a CBAM ahead of the SPP layer, allowing the network to emphasize salient features before performing multi-scale training. The structures of the SPP and CBAM modules are jointly illustrated in Fig. 7.

3.4. *Data Augmentation Strategy*

Human action characterization must account for inherent biological variability. Consider two individuals executing identical motions – one standing 1.8 meters tall, the other 1.6 meters. Their height disparity introduces measurement artifacts. Furthermore, inter-subject kinematic variations manifest in three domains: temporal (movement pacing discrepancies), spatial (viewpoint-dependent morphological distortions), and directional (trajectory orientation effects). These observations motivate our multi-modal augmentation protocol incorporating camera perspective adjustments, anthropometric scaling, and velocity profile warping.

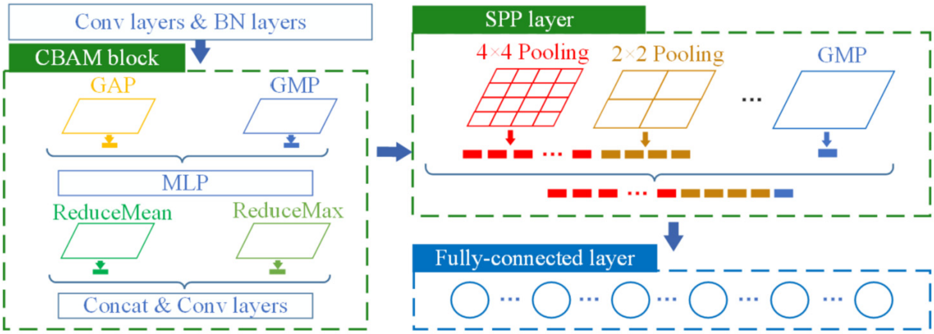


Fig. 7. The overall architecture of the CBAM block and SPP layer integrated in our network.

Considering that our feature construction choices are based on relative coordinates, the KDJMM is unaffected by viewpoint changes. In this experiment, we follow our previous work (Yao *et al.*, 2020). Body size was controlled by multiplying all coordinates by a body size coefficient (BS). The speed of movement is controlled by the velocity coefficient (VS). We control the difference between the body shape and the speed around 0.2, BS = (0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.10, 1.15, 1.2) and VS = (0.8, 0.9, 1.0, 1.1, 1.2, 1.3). By combining the BS and the VS, the set of data will be enlarged up to 54 times of the original one. The diversity of data will be effectively increased, helping the model to accelerate the training process and learn a more comprehensive representation of features, thus improving the accuracy of classification or detection.

The proper setting of VS and BS coefficients is crucial. Taking a 1.7-meter-tall individual as an example, when BS = 1.5 is applied, their height would be scaled to 2.55 meters, which clearly exceeds normal human body dimensions. Similarly, excessively large VS coefficients would distort the speed characteristics of actions like walking and running, making it difficult to distinguish these basic motion patterns. We recommend maintaining these ranges: $0.8 \leqslant \mathrm{BS} \leqslant 1.2$ and $0.8 \leqslant \mathrm{VS} \leqslant 1.3$.

4. Experimental Results

The experiments are based on the use of TensorFlow-gpu 2.7.0 and Keras on a desktop computer equipped with an Nvidia GTX 3060 GPU, an Intel Core i7-11700KF 3.60 GHz processor and 32 GB of RAM clocked at 2666 MHz.

We conduct experiments on two public small datasets, UTKinect-Action3D (UTK-3D) and Florence-3D (FL-3D), and a self-collected HanYue-3D (HY-3D) dataset. We comprehensively validate the effectiveness of keyframe extraction strategies, KDJMM, L-KFNet, as well as additional strategies such as data augmentation, multi-scale learning, and attention mechanisms.

4.1. *Dataset*

The FL-3D dataset was collected in 2012 at the University of Florence, using a Kinect camera, and contains nine activities: "wave, drink from a bottle, answer phone, clap, tie lace, sit down, stand up, read watch, bow.” Ten subjects performed each action two to three times, yielding 215 samples in total, with approximately 20–30 samples per action. For each subject, 15 skeletal joint points were recorded.

The UTK-3D dataset consists of depth sequences collected at 15 fps using a Kinect camera from the Windows SDK Beta version, providing RGB, depth, and 3D skeleton data. The dataset covers ten daily activities: “walking, sitting down, standing up, picking up, carrying, throwing, pushing, pulling, waving and clapping.” Ten subjects performed each action twice, but only 19 samples of the carrying action were available as the skeleton information for one sample was not captured. Twenty joints were recorded for each subject.

The HY-3D dataset was independently collected, using the Kinect v2.0 camera, and covered 15 simple movement types: “talking on the phone, drinking water, waving, looking at the clock, patting dust off clothes, falling, pushing a chair, jumping in place, standing still, clapping, walking, sitting down, sitting still, and sitting and clapping.” Nine subjects were asked to perform each of these 15 activities between three and four times. They recorded the 3D coordinates of all 25 joints provided by the Kinect v2.0 sensor, obtaining 413 samples, with each movement type represented by 35 to 37 samples. In addition, four complex behaviour types were collected to study the timing of the movements, namely “Sit – Stand up – Pat dust on clothes, Jump – Sit – Jump – Wave, Walk – Sit – Jump – Wave – Sit and Sit down and clap – Stand up and clap – Jump – Wave.” Fifteen of these simple movement types were used for our current study.

4.2. *Key Frame Extraction Evaluation*

In this section, we evaluate the extraction of key frames by visualizing the clustering process, comparing the colour distributions of frames in the clusters produced by different clustering methods, and finally obtaining a sparse motion history map (SMHI). As demonstrated in Fig. 8, SMHIs effectively represent motion trajectories for elementary actions, validating the feasibility of algorithmically extracted key frames for HAR.

HAR differs from simple image classification. HAR not only relies on spatial information but also on temporal information. As shown in Fig. 8(b), traditional clustering methods (DBSCAN, HC, SC, K-means) cannot distinguish frames with the same spatial position but opposite velocity directions for periodic actions such as jumping in place in keyframe extraction. In the figure, frames with significant colour differences are clustered together. Figure 8 shows that the proposed method maintains temporal coherence. The cluster colours range from light to dark, and the method decomposes the “jumping in place” action into crouch preparation, take-off, and landing. This result verifies that the proposed method effectively addresses temporal disorder and accurately extracts the key frames of the action.

4.3. *Methodological Assessment and Comparison*

Considering that MHI is widely known in the field of HAR, and that the original MHI and the key frame extracted SMHI behave as the exact size of images, their computational efficiency as input to the model is the same. We used both representations to compare HAR

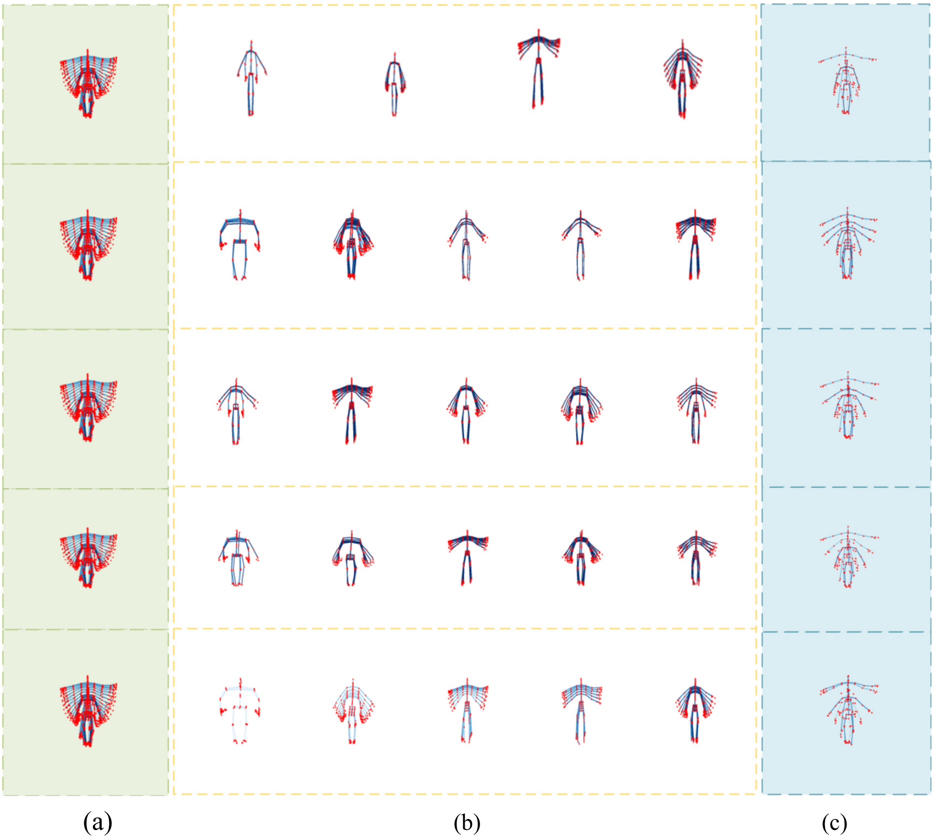


Fig. 8. Visualization of extracted key frames. Taking "jumping in place" as an example, five key frames are extracted. From top to bottom, the clustering processes of DBSCAN, Hierarchical Clustering (HC), Spectral Clustering (SC), K-means, and KKF are shown. (a) shows the classical original Motion History Image (MHI), (b) illustrates the visualization of frames within each cluster, and (c) presents the SMHI generated by the algorithm.

accuracy, thereby evaluating the effectiveness of the key frames. Three scales of SMHI (SMHI-5, SMHI-10, SMHI-15) are extracted using angle and displacement. The accuracy of different clustering methods on the dataset and the time of extracting key frames are compared based on key frames to verify the validity of the algorithm. Then, the validity of KDJMM is demonstrated by comparing the performances of SMHI and KDJMM on the dataset. Throughout the experiment, 80% of the dataset is used for training and 20% for testing. The number of training rounds is set to 200. Meanwhile, an early stopping mechanism is employed to halt training when the loss falls below 0.005, preventing overfitting.

4.3.1. *Evaluation and Comparison of Key Frame Effectiveness*

In the experiments, we first compare in detail the test results of the four data forms on three datasets. The detailed experimental records are presented in Table 1. The main ad-

Table 1
Comparison of MHI and SMHI for different key frames.

Date	Criterion	Accuracy (%)		
		HY-3D	UTK-3D	FL-3D
MHI	ACC	70.12	58.97	77.50
	TOP-3	92.20	76.92	90.00
	TOP-5	96.10	87.17	**100.00**
SMHI+A 5 key frames	ACC	59.74	48.71	72.50
	TOP-3	87.01	84.61	90.00
	TOP-5	96.10	94.87	97.50
SMHI+A 10 key frames	ACC	**76.62**	**64.10**	**82.50**
	TOP-3	**96.10**	**87.17**	**92.50**
	TOP-5	97.40	**100.00**	95.00
SMHI+A 15 key frames	ACC	68.83	**64.10**	77.50
	TOP-3	89.61	84.61	**92.50**
	TOP-5	**98.70**	92.30	**100.00**

Table 2
Comparison of different clustering methods SMHI.

Method	Criterion	Accuracy (%)			Time (s)
		HY-3D	UTK-3D	FL-3D	
DBSCAN	ACC	61.03	48.71	55.00	**9.26**
	TOP-3	88.31	84.61	75.00	
	TOP-5	96.10	94.87	87.50	
HC	ACC	72.72	61.53	**82.50**	4711.87
	TOP-3	92.20	**97.17**	90.00	
	TOP-5	**97.40**	**100**	**97.50**	
SC	ACC	71.42	48.71	75.00	24.87
	TOP-3	92.20	84.61	87.50	
	TOP-5	97.40	94.87	92.50	
K-means	ACC	75.32	**64.10**	77.50	139.36
	TOP-3	93.50	92.30	87.50	
	TOP-5	96.10	97.43	95.00	
KKF	ACC	**76.62**	**64.10**	**82.50**	46.17
	TOP-3	**96.10**	87.17	**92.50**	
	TOP-5	**97.40**	**100.00**	95.00	

vantage of SMHI over MHI is that it can alleviate the spatio-temporal overlap problem of MHI, especially for some actions that contain repetitive behaviours. However, when repetitive behaviours occur very quickly, this problem still remains challenging. As shown in Table 1, the SMHI performance based on our 10-frame extraction method is the best overall. Therefore, the following HAR experiments are based on ten frames. To verify the effectiveness of our proposed algorithm, Table 2 shows the comparison between the key frames extracted by different clustering methods and the SMHI generated by our method. Our method is optimal in terms of accuracy, which verifies the conclusion of 4.2 that the classical clustering algorithms do not take into account the temporal sequence information of human behaviours. Secondly, our algorithm adds more computational metrics to make the key frame selection more precise. Although the DBSCAN algorithm is optimal

Table 3
Comparison of different clustering methods KDJMM.

Dateset	Criterion	Accuracy (%)			
		HC	SC	K-meas	Our method
HY-3D	ACC	79.22	77.92	79.22	**83.11**
	TOP-3	**98.70**	97.40	97.40	**98.70**
	TOP-5	**98.70**	97.40	**98.70**	**98.70**

in terms of computational efficiency, it cannot unify the number of key frames extracted. In practice, the actual effectiveness of the algorithm mainly depends on two parameters required by the algorithm: neighbourhood radius (Eps) and minimum number of neighbours (MinPts). It is often difficult to give a standard value for these two parameters for different behaviours of the human body. In our experiments, we finally set Eps $= 0.5$ and MinPts $= 2$.

Tables 2 and 3 show the time required to extract key frames using different clustering techniques and the recognition accuracy of the constructed SMHI and KDJMM. The proposed keyframe extraction method performs the best in overall performance, while requiring only about 1/100 of the time needed by the HC method (the second most accurate method). The HAR accuracy of KDJMM improves by 4–9% compared to SMHI when used as input. The improvement can be attributed to KDJMM's ability to provide more detailed features, which enables stronger robustness under changes in viewpoint and spatio-temporal overlap of behaviours.

4.3.2. *Evaluation of Adding Key Frame Indicators*

We conduct comparative experiments to verify that adding angle metrics to the algorithm can improve the accuracy of keyframe extraction. In the experiment, we set up two data input forms: KDJMM-DA constructed by combining displacement and angle indicators to extract key frames, and KDJMM-D constructed by only using displacement indicators to extract key frames. All other conditions are kept the same and experiments are conducted on three datasets. As illustrated in Table 4, the results on both the HY-3D and UTK-3D datasets indicate that using the KDJMM-DA data format can achieve higher accuracy. This indicates that introducing appropriate indicators is beneficial for keyframe selection. Notably, the proposed method can adjust these indicators according to specific application scenarios to achieve a balance between accuracy and efficiency.

4.3.3. *Evaluation of L-KFNet*

The scale of KDJMM is relatively small, which means it is not suitable for conventional deep CNNs such as VGG16 and VGG19. We design L-KFNet and compared it with classical neural networks to validate the effectiveness of the proposed network. Detailed experimental records are presented in Table 5 and Fig. 9.

As shown in Table 5, among the listed classical neural networks, our proposed network is optimal in the test accuracy of recognition and the training time. We visualize the network's training process to further validate the effectiveness of the proposed network.

Table 4
Comparison of the accuracy of key frame extraction with and without angle metrics.

Dateset	Criterion	Accuracy (%)		
		HY-3D	UTK-3D	FL-3D
KDJMM-DA	ACC	**83.11**	**89.74**	95.00
	TOP-3	**98.70**	100.00	100.00
	TOP-5	**98.70**	100.00	100.00
KDJMM-D	ACC	80.51	87.17	95.00
	TOP-3	97.40	100.00	100.00
	TOP-5	97.40	100.00	100.00

Table 5
Comparing the training results of different network methods for KDJMM.

Typical CNN	KDJMM					
	Training (%)	ACC (%)	TOP-3 (%)	TOP-5 (%)	Time (s)	MFLOPs
ResNet50	98.80	77.92	89.60	94.80	78.91	63.75
ResNet50V2	**100.00**	76.62	93.50	98.70	69.79	61.60
ResNet101	97.30	70.12	90.90	97.40	142.58	101.65
ResNet101V2	**100.00**	79.22	96.10	98.70	143.70	99.50
ResNet152	91.39	70.12	89.61	98.70	236.35	139.57
ResNet152V2	98.51	75.32	93.50	94.80	230.24	137.40
VGG16	87.83	81.81	97.40	**100.00**	48.93	2074.20
VGG19	87.24	75.32	92.20	97.40	66.63	3176.24
ZFNet	**100.00**	70.12	89.61	97.40	25.42	1118.89
MobileNetV2	**100.00**	76.62	97.40	97.40	6.79	30.03
MobileNetV3Small	99.40	9.09	22.07	35.06	59.92	**3.38**
Xception	**100.00**	70.12	88.31	94.80	20.17	44.66
L-KFNet	**100.00**	**83.11**	**98.70**	98.70	**4.25**	59.55

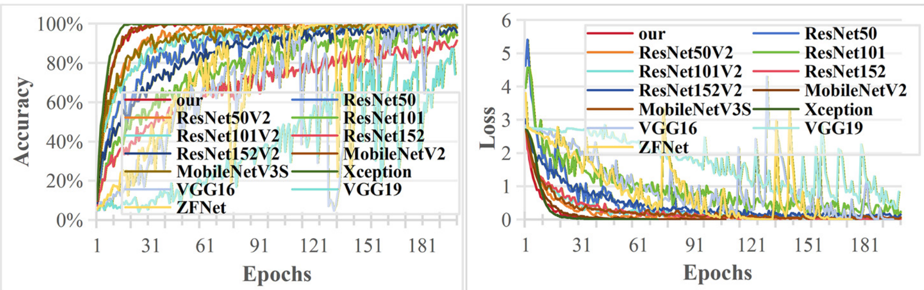


Fig. 9. Accuracy and loss curves for each network training (a) Accuracy curve (b) Loss curve.

In Fig. 9, the proposed network converges quickly and learns discriminative features. This proves it can effectively learn the constructed KDJMM and perform HAR.

Table 6
Comparing the training results of different network methods for KDJMM.

Method	Criterion				
	Accuracy (%)	Precision (%)	Recall (%)	F1-score	Training time (s)
LSTM	70.13	71.67	69.1	0.67	33.94
LSTM+CBAM	83.11	84.49	82.71	0.82	64.04
BiLSTM	75.32	74.76	74.71	0.72	39.18
BiLSTM+CBAM	80.52	80.22	80.76	0.79	51.32
L-KFNet	83.11	84.04	83.05	0.82	**4.25**
L-KFNet+CBAM	**84.41**	**85.93**	**84.38**	**0.84**	5.75

To further highlight the advantages of the proposed L-KFNet over recurrent architectures, we conducted additional comparative experiments with LSTM and BiLSTM, both with and without the CBAM attention mechanism. As shown in Table 6, on the constructed KDJMM, LSTM and BiLSTM still suffer from long-term dependency issues in long-sequence modelling, leading to slower training convergence and limited overall performance. In contrast, the L-KFNet demonstrates superior performance across all evaluation metrics. Furthermore, the CBAM enhances all deep learning models by adaptively refining spatial and channel representations. Notably, L-KFNet+CBAM achieves the best overall balance, with superior recognition accuracy and remarkably low training time compared to recurrent baselines.

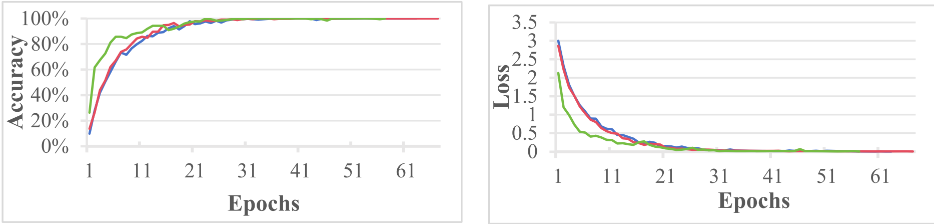
4.3.4. *Evaluation of Multi-Scale, Data Augmentation, and Attention Strategies*

In the preceding experiments, we demonstrate the advantages of our proposed key frame algorithm within clustering algorithms and compared the lightweight network with classical CNNs based on the constructed KDJMM. Nevertheless, the challenge of limited training samples remains. The proposed method first uses the proposed data augmentation strategy to expand the training samples. Then, it extracts different scales of KDJMM from the samples to implicitly expand the training samples.

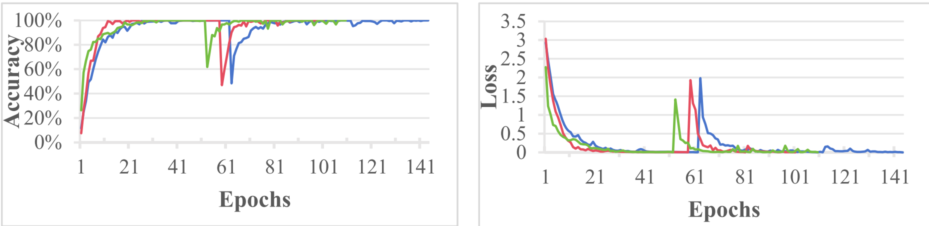
In this experiment, three scales are set: 5, 10, and 15 frames, to construct KDJMMs of different sizes. Notably, the proposed KKF algorithm can easily expand this scale to obtain more temporal information. However, this also leads to increased training time for the model. In addition, we embed an attention module into the lightweight CNN to adaptively focus on discriminative joints and frames, thereby further improving recognition accuracy.

As illustrated in Fig. 10, Fig. 10(a) depicts the training process of the benchmark (N), Fig. 10(b) portrays the training process with multi-scale (MS), Fig. 10(c) illustrates the training process with data augmentation (DA), Fig. 10(d) demonstrates the training process with multi-scale plus data augmentation (MS+DA), and Fig. 10(e) demonstrates the training process with the combination of multi-scale, data augmentation, and attention mechanism (MS+DA+AM). In addition, the quantitative results of these strategies on three benchmark datasets are summarized in Table 7, further validating the effectiveness of combining multi-scale, data augmentation, and attention mechanism.

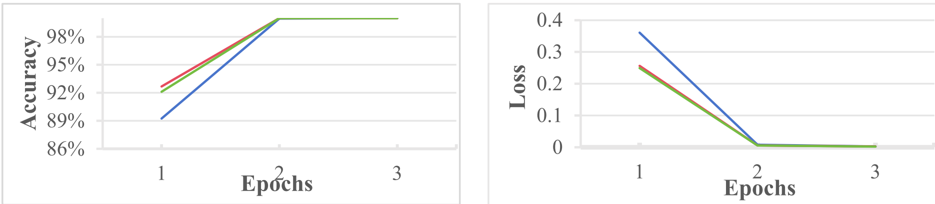
As shown in Fig. 10, when L-KFNet combines multi-scale learning, data augmentation, and attention mechanisms, the accuracy of HAR improves across all three datasets.



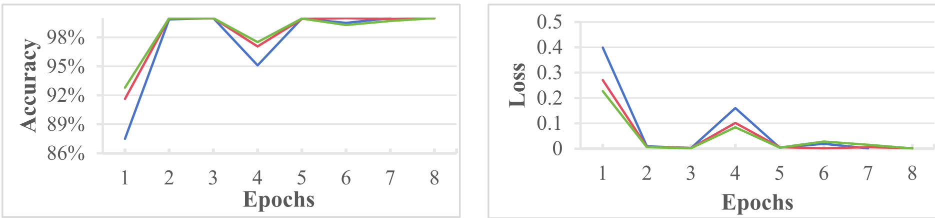
(a)



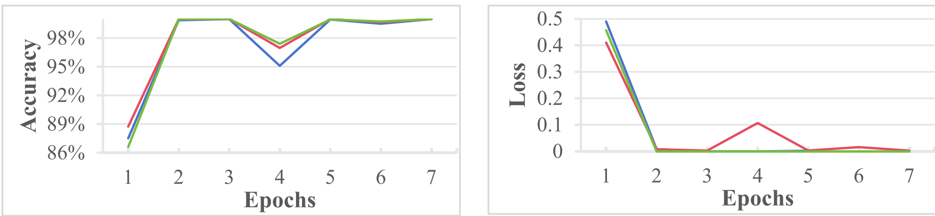
(b)



(c)

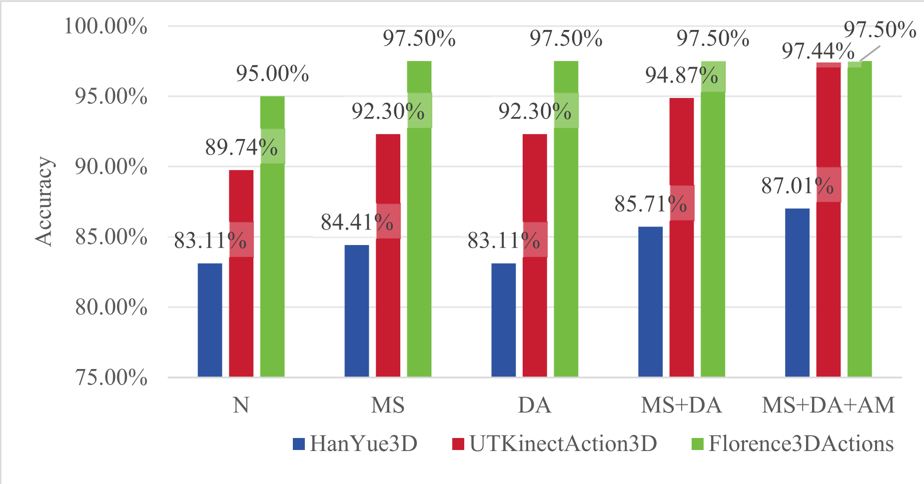


(d)



(e)

Fig. 10. (a)–(e) illustrates the accuracy and loss curves of the training process based on the three datasets of HY-3D, UTK-3D, and FL-3D with varying training strategies. (f) depicts the corresponding histogram of the test results.



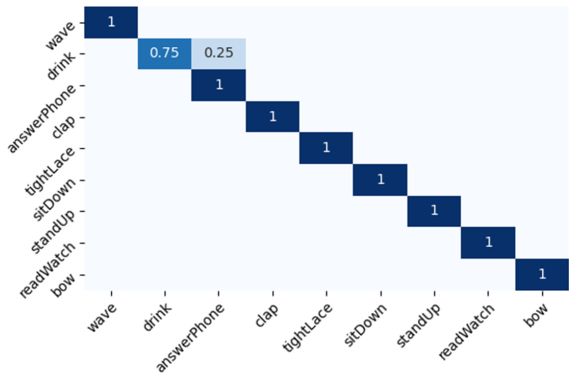
(f)

Fig. 10. (*continued*)

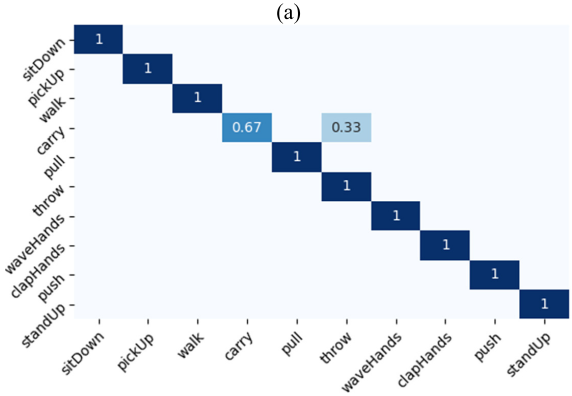
Table 7
Performance comparison of different strategies on three benchmark datasets.

Method	Dataset	Criterion			
		Accuracy (%)	Precision (%)	Recall (%)	F1-score
N	HY-3D	83.11	84.04	83.05	0.82
	UTK-3D	89.74	90.50	90.00	0.89
	FL-3D	95.00	95.93	94.44	0.94
MS	HY-3D	84.41	85.04	84.05	0.83
	UTK-3D	92.30	93.50	92.50	0.91
	FL-3D	**97.50**	**97.78**	**97.22**	**0.97**
DA	HY-3D	83.11	83.98	83.05	0.82
	UTK-3D	92.30	93.50	92.50	0.91
	FL-3D	**97.50**	**97.78**	**97.22**	**0.97**
MS+DA	HY-3D	85.71	86.47	85.71	0.84
	UTK-3D	94.87	95.50	95.00	0.94
	FL-3D	**97.50**	**97.78**	**97.22**	**0.97**
MS+DA+AM	HY-3D	**87.01**	**88.19**	**87.05**	**0.86**
	UTK-3D	**97.44**	**98.00**	**96.67**	**0.96**
	FL-3D	**97.50**	**97.78**	**97.22**	**0.97**

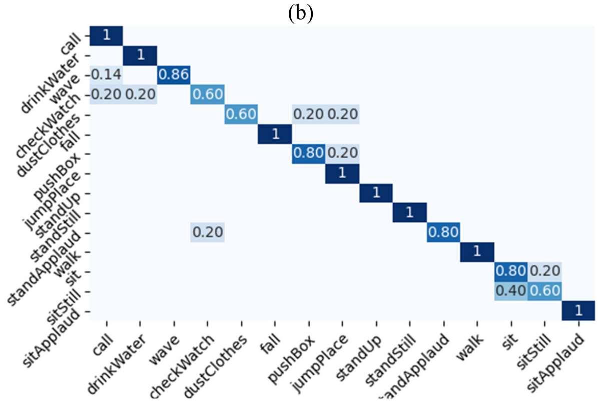
Specifically, by combining L-KFNet with data augmentation, the accuracy of the UTK-3D dataset increases by 2.56%, and the FL-3D dataset increases by 2.5%. After further combining multi-scale training, the accuracy of the HY-3D dataset increases by 1.3%. After introducing the attention mechanism, the accuracy further improves to 97.44% for UTK-3D and 87.01% for HY-3D, and 97.5% for FL-3D. The experimental results are shown in Table 7.



(a)



(b)



(c)

Fig. 11. Confusion matrices (a–c) obtained by our proposed method on the three datasets FL-3D, UTK-3D and HY-3D.

4.4. *Analysis of Results*

We evaluated our proposed method on three datasets. The experimental results of confusion matrices (Fig. 11) and overall accuracy (Table 8) demonstrated its effectiveness.

Table 8
Comparison of previous state-of-the-art methods with ours.

Method	Acc (%)		
	FL-3D	UTK-3D	HY-3D
DJMI + ZfNet + DA	92.50	91.84	–
DJMI + ZfNet + LSTM + DA[29]	93.77	94.23	–
MS + ZfNet + DA[23]	94.74	94.74	83.87
MHI + L-KFNet	77.5	58.97	70.12
SMHI + L-KFNet	82.50	64.10	76.62
KDJMM-D + L-KFNet	95.00	87.17	80.51
KDJMM-DA + L-KFNet	95.00	89.74	83.11
KDJMM-DA + L-KFNet + DA	**97.50**	92.30	83.11
KDJMM-DA + L-KFNet + MS	**97.50**	92.30	84.41
KDJMM-DA + L-KFNet + DM	**97.50**	92.30	84.41
KDJMM-DA + L-KFNet + MS + DA	**97.50**	94.87	85.71
Our method	**97.50**	**97.44**	**87.01**

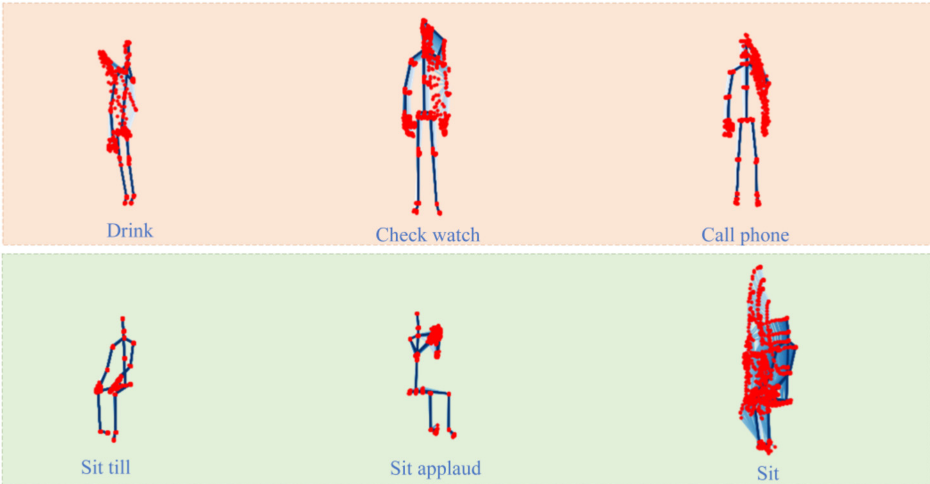


Fig. 12. HanYue-3D dataset similarity actions.

Specifically, the confusion matrices for FL-3D and UTK-3D (Fig. 11a–b) show that our method not only classifies different action categories such as “clap, bow, and sit down”, but also correctly classifies most of the similar action categories, such as “stand up, sit down, and stand still”. Only the actions of “drink and carry” exhibited slightly lower recognition rates. However, the recognition accuracy of sit-still and check watch behaviours remains limited in the HY-3D dataset. The main reason is that there are many similar actions in HY-3D, such as “drink, check watch, call phone” or some composite actions contain some other atomic actions, such as “sit, sit still, sit-applaud”. These may bring ambiguity to our proposed method. To further investigate the cause of these confusions, we visualized some similar actions in the HY-3D dataset (Fig. 12). The actions like “drinking water, looking at a watch, making a phone call” are all share a highly similar arm-raising gesture towards the head. Similarly, all the actions of "sitting, sitting still, sitting and clapping" contains the action of sitting. Although the HY-3D dataset provides 25 joints per frame, our current model uses only 15 major body joints. Therefore, incorporating these additional joints into the model's representation could provide distinctive features to mitigate these recognition errors.

Finally, comparative experiments with benchmark methods are conducted to objectively assess our method's performance. The results in Table 8 show that our method achieves superior accuracy across all three datasets. At the same time, our method relies only on key frames for HAR, and it reduces processing time significantly.

5. Discussion and Conclusion

Inspired by animation and film production, this paper introduces a key frame-based method for HAR that avoids processing every video frame. The method significantly reduces computational costs (625 FPS processing speed), avoids the problem of temporal disorder in traditional clustering, and enables normalization of variable-duration actions. Based on the extracted key frames, we derive rich features to construct KDJMM, which is then used in combination with a lightweight CNN. Experiments show that the proposed method improves the recognition efficiency without compromising accuracy. Meanwhile, the combined strategy of data augmentation, multi-scale training, and attention mechanism maintains robustness under limited sample conditions. Although the proposed method performs well in terms of latency and accuracy, there remains room for improving the accuracy of feature representation and key frame extraction. These will be the focus of our future work in HAR.

Conflict of Interest Statement

No author associated with this paper has disclosed any potential or pertinent conflicts that may be perceived to have an impending conflict with this work.

Author Statement

Leiyue Yao: Method, Supervision, Reviewing. **Chao Zeng:** Paper writing, Software, Data Visualization. **Jianying Xiong:** Supervision, Reviewing and Editing, Data Collecting, **Keyun Xiong and Lei Zhang:** Funding. **Yucheng Wang:** Editing.

Funding

This research was supported by the National Natural Science Foundation of China under Grant 62366023, the Scientific and Technological Projects of the Nanchang Science and Technology Bureau under Grant GJJ2202613.

References

Alsaadi, M., Keshta, I., Ramesh, J.V.N., Nimma, D., Shabaz, M., Pathak, N., Singh, P.P., Kiyosov, S., Soni, M. (2025). Logical reasoning for human activity recognition based on multisource data from wearable device. *Scientific Reports*, 15(1), 380.

Biswal, A., Panigrahi, C.R., Behera, A., Nanda, S., Weng, T.H., Pati, B., Malu, C. (2024). Activity recognition for elderly care using genetic search. *Computer Science and Information Systems*, 21(1), 95–116.

Caetano, C., Sena, J., Brémond, F., Dos Santos, J.A., Schwartz, W.R. (2019). Skelemotion: a new representation of skeleton joint sequences based on motion information for 3d action recognition. In: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, pp. 1–8.

Chen, Y.X., Song, Y.H., Huo, F.Z. (2023). Simulation of crowd evacuation behaviours at subway stations under panic emotion. *International Journal of Simulation Modelling*, 22(4), 667–678.

Chen, H., Pan, Y., Wang, C. (2024). An optimization method of human skeleton keyframes selection for action recognition. *Complex & Intelligent Systems*, 10(4), 4659–4673.

Du, Y., Fu, Y., Wang, L. (2015). November. Skeleton based action recognition with convolutional neural network. In: *2015 3rd IAPR Asian Conference on Pattern Recognition*. IEEE, pp. 579–583.

Dutta, S.J., Boongoen, T., Zwiggelaar, R. (2025). Human activity recognition: a review of deep learning-based methods. *IET Computer Vision*, 19(1), e70003.

Gan, M., Liu, J., He, Y., Chen, A., Ma, Q. (2023). Keyframe selection via deep reinforcement learning for skeleton-based gesture recognition. *IEEE Robotics and Automation Letters*, 8(11), 7807–7814.

Kar, T., Kanungo, P., Mohanty, S.N., Groppe, S., Groppe, J. (2024). Video shot-boundary detection: issues, challenges and solutions. *Artificial Intelligence Review*, 57(4), 104.

Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F. (2017). A new representation of skeleton sequences for 3d action recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3288–3297.

Khezerlou, F., Baradarani, A., Balafar, M.A. (2023). A convolutional autoencoder model with weighted multi-scale attention modules for 3D skeleton-based action recognition. *Journal of Visual Communication and Image Representation*, 92, 103781.

Kong, Y., Fu, Y. (2022). Human action recognition and prediction: a survey. *International Journal of Computer Vision*, 130(5), 1366–1401.

Lanzoni, D., Harih, G., Buchmeister, B., Vujica-Herzog, N. (2024). Process simulate versus inertial Mocap system in human movement evaluation. *International Journal of Simulation Modelling*, 23(4), 587–598.

Li, X., Kang, J., Yang, Y., Zhao, F. (2023). A lightweight attentional shift graph convolutional network for skeleton-based action recognition. *International Journal of Computers Communications & Control*, 18(3), 5061.

Liu, J., Akhtar, N., Mian, A. (2019). Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 10–19.

Liu, Q., Yun, F., Dong, M., Djoric, D., Zivlak, N. (2024). Health prognosis for equipment based on ACO-K-means and MCS-SVM under small sample noise unbalanced data. *Tehnički vjesnik*, 31(1), 24–31.

Phan, H.H., Nguyen, T.T., Phuc, N.H., Nhan, N.H., Tran, C.T., Vi, B.N. (2021). Key frame and skeleton extraction for deep learning-based human action recognition. In: *2021 RIVF International Conference on Computing and Communication Technologies*. IEEE, pp. 1–6.

Simonyan, K., Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, *27*.

Xin, W., Liu, R., Liu, Y., Chen, Y., Yu, W., Miao, Q. (2023). Transformer for skeleton-based action recognition: a review of recent advances. *Neurocomputing*, 537, 164–186.

Yang, Z., Li, Y., Yang, J., Luo, J. (2018). Action recognition with spatio–temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8), 2405–2415.

Yao, L., Yang, W., Huang, W. (2020). A data augmentation method for human action recognition using dense joint motion images. *Applied Soft Computing*, 97, 106713.

Yao, L., Yang, W., Huang, W., Jiang, N., Zhou, B. (2022). Multi-scale feature learning and temporal probing strategy for one-stage temporal action localization. *International Journal of Intelligent Systems*, 37(7), 4092–4112.

Yu, X., Zhang, X., Xu, C., Ou, L. (2024). Human-robot collaborative interaction with human perception and action recognition. *Neurocomputing*, 563, 126827.

Zhang, F.L. (2024). Evolutionary algorithm for dynamic resource allocation and its applications. *International Journal of Simulation Modelling*, 23(3), 531–542.

Zhang, Y., Zhang, J., Liu, R., Zhu, P., Liu, Y. (2023). Key frame extraction based on quaternion Fourier transform with multiple features fusion. *Expert Systems with Applications*, 216, 119467.

Zhao, Y., Guo, H., Gao, L., Wang, H., Zheng, J., Zhang, K., Zheng, Y. (2023). Multi feature fusion action recognition based on key frames. *Concurrency and Computation: Practice and Experience*, 35(21), e6137.

Zhao, Z., Chen, Z., Li, J., Wang, X., Xie, X., Huang, L., Zhang, W., Shi, G. (2024). Glimpse and zoom: Spatio-temporal focused dynamic network for skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7), 5616–5629.

Zhao, Z., Chai, W., Hao, S., Hu, W., Wang, G., Cao, S., Song, M., Hwang, J.-N., Wang, G. (2025). A survey of deep learning in sports applications: Perception, comprehension, and decision. *IEEE Transactions on Visualization and Computer Graphics*, *31*.

Zhu, H., Zheng, Z., Nevatia, R. (2023). Gait recognition using 3-d human body shape inference. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 909–918.

L. Yao received the BE, ME, and PhD degrees in computer science from Nanchang University, China. He is currently a professor at the School of Intelligent Medicine and Information Engineering, Jiangxi University of Chinese Medicine. He has published several papers in international journals and conferences. His current research interests include vision-based human action recognition, image and video processing, massive data processing, distributed systems, and software engineering.

C. Zeng received the BE degree in network engineering from Hunan Institute of Technology, China, in 2023, and is currently pursuing the MS degree in Computer Science and Technology at Jiangxi University of Chinese Medicine. His research interests include key-frame extraction for human action recognition and vision-based behavior analysis.

J. Xiong is an associate professor in the field of computer science. She received the ME degree from Zhejiang University of Technology, China, in 2006, and the PhD degree from Jiangxi University of Finance and Economics, China, in 2013. Her research interests include information systems, information management, and service computing.

K. Xiong was born in September 1980 in Nanchang, China. He received the BE degree and is currently a lecturer. His research interests mainly include big data architecture and data mining.

L. Zhang received the BS degree in mechanical design and automation from China University of Petroleum (Beijing), China. He is currently a senior engineer in mechanical design and serves as the Deputy General Manager of R&D at Beijing Hanlin Hangyu Technology Development Co., Ltd. His research interests include pharmaceutical technology and the development of intelligent equipment based on computer vision and related technologies.

Y. Wang received the BS degree in computing and software systems from the University of Melbourne, Australia. His research interests include computer vision, intelligent systems, and data-driven applications.