electronics

MDPI

Article

Human Action Recognition Using Key-Frame Attention-Based LSTM Networks

Changxuan Yang [1], Feng Mei [1], Tuo Zang [1], Jianfeng Tu [1], Nan Jiang [1] and Lingfeng Liu [1,2,*]

[1] School of Information Engineering, East China JiaoTong University, Nanchang 330013, China; changxuan.yang@ecjtu.edu.cn (C.Y.); mei_feng3@163.com (F.M.); tuo.zang@ecjtu.edu.cn (T.Z.); jianfeng.tu@ecjtu.edu.cn (J.T.); jiangnan@ecjtu.edu.cn (N.J.)
[2] Jiangxi Minxuan Intelligent Technology Co., Ltd., Nanchang 330029, China
* Correspondence: lingfeng.liu@ecjtu.edu.cn

Abstract: Human action recognition is a classical problem in computer vision and machine learning, and the task of effectively and efficiently recognising human actions is a concern for researchers. In this paper, we propose a key-frame-based approach to human action recognition. First, we designed a key-frame attention-based LSTM network (KF-LSTM) using the attention mechanism, which can be combined with LSTM to effectively recognise human action sequences by assigning different weight scale values to give more attention to key frames. In addition, we designed a new key-frame extraction method by combining an automatic segmentation model based on the autoregressive moving average (ARMA) algorithm and the K-means clustering algorithm. This method effectively avoids the possibility of inter-frame confusion in the temporal sequence of key frames of different actions and ensures that the subsequent human action recognition task proceeds smoothly. The dataset used in the experiments was acquired with an IMU sensor-based motion capture device, and we separately extracted the motion features of each joint using a manual method and then performed collective inference.

Keywords: action recognition; ARMA; attention mechanism; key-frame extraction; K-means; LSTM

Citation: Yang, C.; Mei, F.; Zang, T.; Tu, J.; Jiang, N.; Liu, L. Human Action Recognition Using Key-Frame Attention-Based LSTM Networks. *Electronics* **2023**, *12*, 2622. https://doi.org/10.3390/electronics12122622

Academic Editors: Zbigniew Leonowicz and Michał Jasiński

Received: 13 May 2023
Revised: 4 June 2023
Accepted: 7 June 2023
Published: 10 June 2023

1. Introduction

In recent years, research on human action recognition has developed by leaps and bounds and is now used in various fields, such as video surveillance, intelligent medical care, human–machine collaboration, and intelligent human–machine interfaces [1–4]. This also means that there are increasingly higher requirements for human action recognition algorithms in terms of performance, which is a classic and challenging topic in computer vision research. To date, many methods based on hand-crafted feature representations have been widely used for action recognition due to their advantages, such as simplicity and robustness [5–7]. However, due to the limitations of human cognitive abilities, the method is often database-oriented and difficult to apply to real-life scenarios.

With the development of deep learning techniques, deep learning algorithms have more advantages in the field of human motion recognition than traditional methods [8]. Currently, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are frequently used in the field of human motion recognition. The 3D CNN [9] is a typical algorithm studied in human action recognition tasks. In that work, 3D convolutions are employed to extract features from the spatial and temporal dimensions of video data. This works well for capturing spatial information and has a better performance in image recognition at the moment, but temporal information is inevitably lost when sequences are encoded into images, and temporal motion plays a key role in human action recognition. This problem can be mitigated with RNNs, in particular long short-term memory (LSTM), which has been shown to effectively model long-term cues of motion sequences [10]. The gate unit in LSTM can choose whether to update specific information or not while ensuring

long-term memory of valid data and forgetting or discarding useless information, thereby maximizing the utilization of data information.

Based on these facts, the accurate recognition of actions in the real world remains challenging. Current human action recognition methods solve the sequence learning problem with LSTM and gated recurrent units but do not focus on the selective information in the sequence in the selection of features. In a human action sequence, not all frames in the sequence are equally important, and there are often repetitive or redundant frames [11] that are not so important for the recognition of the action. To solve this problem, some researchers [12] introduced a key-frame mechanism to increase the information difference among motion frames. The key frames of a motion sequence are extracted using clustering, and the redundant frames of the motion sequence are discarded to reconstruct the motion. The method can effectively improve the recognition efficiency of the model.

Therefore, in this work, we combine the above problems and propose a human action recognition method based on LSTM with key-frame attention. The key frames of different actions are extracted using clustering. They are combined with attention mechanisms to further improve the recognition efficiency and accuracy of LSTM networks. However, most key-frame extraction methods do not take into account the time-series information between frames [13]. When confronted with long-term motion sequences containing multiple actions, there are often large errors in the localisation of action intervals [14]. This prevents the uniform processing of inter-class similarities in different actions of the motion sequence and tends to cause inter-frame confusion in the timing of key frames of different actions. This has an impact on the accuracy of subsequent action recognition tasks.

To this end, we combine our previous work [15] and propose a new method for key-frame extraction. The automatic segmentation model based on the autoregressive moving average (ARMA) algorithm was combined with the K-means clustering algorithm. This method allows the automatic segmentation of different movements in a motion sequence to be performed in advance. The possibility of inter-frame confusion in the timing of key frames of different actions is effectively avoided. This method ensures that subsequent human movement recognition tasks are carried out smoothly.

Combining these two modules, we propose a key-frame-based human action recognition system. In this work, we trained the model using a 3D skeleton sequence reconstructed from MoCap (motion capture device) data based on IMU sensors. In the recognition of human movement types, 18 representative joints throughout the human body were selected to build the skeleton model of the human body. The motion sequence was constructed from the extracted human pose feature vectors.

The main innovations and contributions of the present study are as follows:

1. We propose a human action recognition model with an LSTM network based on the key-frame attention mechanism. The issues of the accuracy and efficiency of the recognition model are fully considered. The recognition performance of the LSTM network is effectively improved by the attention mechanism combined with the key frames of the action.
2. We propose an unsupervised learning-based [16] key-frame extraction method. This method can accurately distinguish and extract key frames that can represent different action types from a long motion sequence containing multiple complex action types.

The rest of the paper is structured as follows: Section 2 provides related work on human action recognition. Section 3 describes the structure of the human action recognition method proposed in this paper and its associated components. Section 4 evaluates and compares the recognition accuracy of this paper's model with that of other human action recognition-related models. Finally, Section 5 draws conclusions.

2. Related Work

In the field of artificial intelligence, human action recognition is an important part of research in this area, making human interaction with the external environment possible. While human communication can be conveyed with words, facial expressions, written text,

etc., the relationship between computers and sensors to understand human intentions and behaviour is now a popular area of research. As a result, more and more researchers are devoting their time and experience to the study of human action recognition.

2.1. Traditional Machine Learning and Hand-Crafted Feature-Based Action Recognition

In traditional recognition methods based on machine learning and manual features, hand-crafted feature extractors and action classifiers based on traditional machine learning algorithms are often used [17]. Action classifiers are used to recognise and classify human movement actions based on the characteristics of that action. For example, Cho et al. [18] used joint distance features for feature extraction. The category of each pose is labelled by an artificial neural network (ANN). Finally, discrete Hidden Markov Models (HMMs) are applied to classify and recognise action sequences. Meanwhile, in order to effectively improve the recognition performance of the system, some researchers have adopted a key-frame-based approach to reduce the processing time of the system [19,20]. A recognition system for human action sequences was developed using traditional machine learning algorithms combined with key-frame selection. In past research, action recognition methods based on traditional machine learning and manual features were combined with great success. However, for the construction and extraction of features [21], they need to rely on human cognition. Moreover, based on human expertise, only superficial features can be learned, making it difficult to cope with the needs of real environments.

2.2. Deep Learning-Based Action Recognition

In recent years, a number of new methods have been developed, especially regarding the application of deep learning methods in action recognition [22]. The main representative works can be summarized as discussion methods based on convolutional neural networks and discussion methods based on LSTM.

Traditional CNN models are currently limited to processing 2D inputs and are not suitable for the feature capture of 3D skeleton data. To shift CNNs from images to temporal motion sequences, Tran et al. [23] extended traditional CNNs to 3D CNNs, which are more suitable for spatio-temporal feature learning. Related experiments have shown that this scheme outperforms traditional 2D CNNs in terms of analytical functionality. Another common strategy is to employ two-stream CNNs to deal with the problem of capturing motion information between consecutive frames. Zhu et al. [24] proposed a CNN architecture based on a two-stream approach that implicitly captures motion information between adjacent frames and uses an end-to-end CNN approach to learn optical streams. Task-specific motion representations can be obtained while avoiding expensive computation and storage. Since then, many improved models have been proposed, and the two-stream CNN has made significant contributions to the development of motor action recognition [25]. It can even be referenced to realistic and complex real-world environments; for example, Hu et al. [26] introduced a video triple model to obtain additional timestamp information, thus extending behaviour recognition to workflow recognition. Moreover, with extensive simulation experiments, it was shown that the algorithm is robust and efficient in the recognition of real environments.

However, these algorithms have been shown to be only effective for short-term temporal feature learning and are not applicable to long-term temporal feature encoding. With the development of RNNs, LSTM networks suitable for long-term motion sequences have been developed. They have been gradually applied to human action recognition, demonstrating their ability to effectively alleviate the recognition problem of long-term motion sequences [27,28]. Wang et al. [29] introduced long short-term memory (LSTM) to model the high-level temporal features generated by a kinetically pretrained 3D CNN model, with satisfactory results in the recognition and classification of long-term motion sequences. However, the traditional frame-skipping pattern of LSTM [30] also limits performance in action recognition. The problem of data redundancy accompanies the task of the recognition of long-term motion action data.

2.3. Action Recognition Based on Joint-Aware and Attention Mechanisms

In recent years, many researchers have turned their attention to joint-aware and attention mechanisms and have achieved good recognition performance in long-term temporal reasoning tasks. Regarding joint-aware-based recognition methods, Oikonomou et al. [31] argue that each action in real life can be effectively perceived by observing only a specific set of joints and associate a specific joint with each action to point out the joint that contributes the most; Shah et al. [32] separately extracted the motion features of each joint using a motion encoder and then performed collective reasoning and selected the most discriminative joint for the recognition task. Regarding recognition methods based on attention mechanisms, Dai et al. [30] proposed an LSTM network based on end-to-end two-stream attention, which can selectively focus on the effective features of the original input image and give different levels of attention to the output of each depth feature map to effectively improve the recognition performance of the model by adopting a visual attention mechanism to address the problem that features of different frames have different learning roles; Li et al. [33] proposed a spatio-temporal attention (STA) network to learn discriminative feature representations of actions by representing useful information at the frame level and channel level, which can be inserted into state-of-the-art 3D CNN architectures for video action detection and recognition with better recognition performance; in the article [34], the authors proposed a bi-directional long short-term memory (BiLSTM)-based attention mechanism. The attention mechanism is used to improve performance and extract additional high-level selective action-related patterns and cues, thereby obtaining a high-performance recognition model.

3. Keyframe-Based Human Action Recognition Method

We propose a key-frame-based human action recognition method, which consists of two main components: a key-frame extraction method and a recognition model:

1. Unsupervised learning-based key-frame extraction method. An unsupervised segmentation model based on the ARMA algorithm is used to automatically segment complex motion sequences containing multiple actions into multiple sub-motion sequences. The K-means clustering algorithm is then used to separately extract the key motion frames from the segmented sub-action sample sequences. Finally, the human action pose feature matrix is labelled and encoded by combining the temporal features of the human motion sequences.
2. Selection and construction of recognition models. In this chapter, three models, HMM, 3D CNN, and LSTM, are employed for the recognition task, and the LSTM network is improved by combining the attention mechanism and by designing an LSTM network based on the key-frame attention mechanism, which further optimises the LSTM network by redistributing the weights of the motion feature sequences using the attention method based on key action frames. The network structure is further optimised to improve the accuracy of human action recognition.

3.1. Construction of Skeletal Models and Feature Sequences

The primary task of recognition algorithms for human movements is to collect and process the motion data of different behavioural actions. During human limb movements, the angular information and spatial location information of the limb bones can be obtained according to the different semantics and postures of the movements.

3.1.1. Structural Characterisation of the Human Body

The motion data collected using a MoCap device was processed in this study in a manner consistent with previous work [15]. As shown in Figure 1 [15], the human skeleton model is a tree-like hierarchical model that consists of a root node and multiple subtrees. The entire skeleton model can be roughly divided into 18 bone segments, each of which has a parent segment and a number of subsegments, with the parent segment and subsegments being connected by joints. When the human body moves, the movement of individual limbs

can be described as the movement of the bone segment of that limb relative to the joints of its parent segment; human limbs periodically switch between flexed and extended postures, and the limbs then show periodic changes that form correlations among limbs. For this reason, limb segmentation angles are introduced to improve the semantic description of motion sequences. In human model building, the hip node is usually chosen as the root node of a tree human skeleton model, which constrains its children. The skeleton model is represented by the coordinates of the spatial position of each joint point; therefore, data on the rotation angle of each joint point need to be converted to the coordinates of the joint point.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = M_r * \left(M_{r-1} * \left(\ldots * \left(M_2 * \left(M_1 * \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \right) \right) \right) \right), \tag{1}$$

$$P = P_{root} + O_{r-1} + \ldots + O_2 + O_1 + O_0, \tag{2}$$

where M_r is the rotation matrix of the joint point, P_{root} is the location of the root node, and O_r is the position of the child node relative to the parent node.

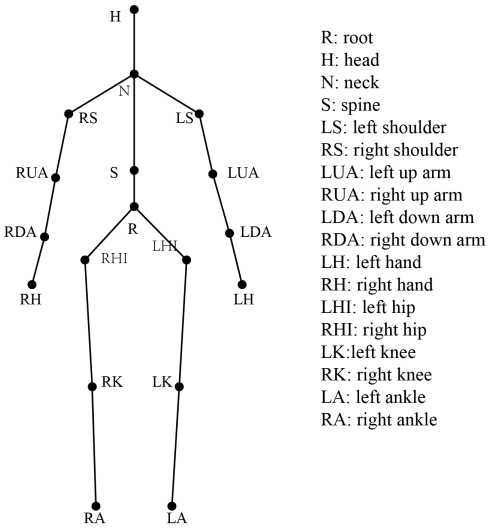


Figure 1. Basic human skeleton structure.

3.1.2. Construction of Feature Sequences

When the human body moves, the movement of each limb can be described as the movement of that limb segment relative to the joints of its parent segment. In human body modelling, the hip node is usually chosen as the root node of a tree human skeleton model, which constrains its child nodes.

Before we can perform motion analysis, we need to construct the feature matrix. The feature matrix is a prerequisite in machine learning and data analysis. It is a matrix made up of the fusion of several features that contain key information of the data, with each row representing a sample and each column representing a feature, and is one of the important steps in machine learning and data analysis. In Tables 1–3, the calculation of each of the three parts of a feature is expressed.

Table 1. Calculation of angle characteristics between adjacent bone segments.

Lower Limbs	Upper Limbs
$\grave{}_{ab}$:RHI to RK→RK to RA	$\grave{}_{ef}$:RUA to RDA→RDA to RH
$\grave{}_{cd}$:LHI to LK→LK to LA	$\grave{}_{gh}$:LUA to LDA→LDA to LH

Table 2. Calculation of the angle characteristics between limb bone segments and the central bone segment.

Lower Limbs	Upper Limbs
$\grave{}_a$:RHI to RK→Central	$\grave{}_e$:RUA to RDA→Central
$\grave{}_b$:RK to RA→Central	$\grave{}_f$:RDA to RH→Central
$\grave{}_c$:LHI to LK→Central	$\grave{}_g$:LUA to LDA→Central
$\grave{}_d$:LK to LA→Central	$\grave{}_h$:LDA to LH →Central
Central:R →S	

Table 3. Calculation of the spatial distance characteristics from the midpoint of each bone segment to the central node (hip node).

Lower Limbs	Upper Limbs
d_1:Midpoint (RHI to RK)→Central	d_5:Midpoint (RUA to RDA)→Central
d_2:Midpoint (RK to RA)→Central	d_6:Midpoint (RDA to RH)→Central
d_3:Midpoint (LHI to LK)→Central	d_7:Midpoint (LUA to LDA)→Central
d_4:Midpoint (LK to LA)→Central	d_8:Midpoint (LDA to LH) →Central
Central node:R	

The angular characteristics of the movement of the different bone segments are determined by the variation in the size of the angle between the individual bone segments. The calculation of the size of the angle between the bone segments of the limbs and the central bone segment, and that of the size of the angle between adjacent bone segments of the limbs are as follows:

$$\begin{aligned} \theta_{AB} &=< \theta_A, \theta_B >= arc\cos(\frac{\theta_A * \theta_B}{|\theta_A||\theta_B|}), \\ \theta_B &=< \theta_i, \theta_j >= arc\cos(\frac{\theta_i * \theta_j}{|\theta_i||\theta_j|}), \end{aligned} \tag{3}$$

where $\theta \in [0, 180^\circ]$, θ_A is the direction vector on the central spinal bone segment partition, and $\theta_B = \{\theta_a, \theta_b, \ldots, \theta_h\}$ is the direction vector on each limb partition of the body.

The three-dimensional spatial characteristics of the movement of the different bone segments are determined by the variation in the magnitude of the spatial position distance between the individual limb bone segments and the central node. The calculation of the spatial position distances between the nodes is as follows:

$$d = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}, \tag{4}$$

where $X_i = (x_i, y_i, z_i)$ and $X_j = \left(x_j, y_j, z_j\right)$ are the 3D spatial position coordinates of the central node of the human skeleton (hip node) and the 3D spatial position coordinates of each limb bone segment in the Cartesian coordinate system, respectively.

3.2. Key-Frame Extraction Method Based on Unsupervised Learning Model

The key-frame extraction technique [35] is a technique used to extract the most informative frames and eliminate pose redundancy. Various types of motion capture devices often use high sampling frequencies for sampling human motion pose data, with the accompanying disadvantage of generating a large number of repetitive redundant frames when data are collected for certain simple movements. This also has a negative impact on the efficiency of the execution of the subsequent model processing part. Therefore, it is essential to extract key frames from human motion data.

3.2.1. Segmentation of Human Limb Movement Sequences

In our previous work [15], we proposed an unsupervised segmentation algorithm based on the structural representation of the angle between limb bone segments and the fitting of an autoregressive moving average (ARMA) model. We combined the predictive and fitting properties of the ARMA model in time series with the regularity of human motion in time series. Temporal inflection points in human motion sequences are calculated, and inflection points are identified and extracted using an adaptation algorithm to achieve motion sequence segmentation. The method overcomes the limitation that the ARMA model is only applicable to short-term sequence prediction and allows the ARMA model to perform long-motion-sequence segmentation.

Firstly, regarding the ARMA model, it is an important model for the study of time series. It consists of an autoregressive (*AR*) model and a moving average (*MA*) model. In the ARMA model, the data of variable Y_t at any time t is represented as its observation sequence $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$ and historical random disturbance sequence $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots$. The linear combination of ε_{t-q}.. ARMA(p, q) is given in Equation (5).

$$\begin{aligned} Y_t &= AR + MA, \\ AR &= c + \beta_1 Y_{t-1} + \beta_1 Y_{t-2} + \ldots + \beta_p Y_{t-p}, \\ MA &= \lambda_1 \varepsilon_t + \lambda_2 \varepsilon_{t-2} + \ldots + \lambda_q \varepsilon_{t-q} + c, \end{aligned} \tag{5}$$

where p and q are the orders of AR and MA, respectively; β_p and λ_q are the calculated coefficients of AR and MA, respectively; and c is the residual constant.

Secondly, the angle characteristics between each limb bone segment and the central spinal bone segment in the human limb movement sequence are combined in the ARMA model [15]. The ARMA model for clip angle series is represented by Equation (6).

$$\theta_{i_t} = \beta_{i_0} + \beta_{i_1} \theta_{i_{(t-1)}} + \beta_{i_2} \theta_{i_{(t-2)}} + \ldots + \beta_{i_n} \theta_{i_{(t-n)}} + Z_{i_t}, \tag{6}$$

where θ_i are the fitted data of the limb bone segment angle, β_{i_n} is the linear approximation factor, and Z_{i_t} is the residual.

After ARMA model fitting is completed for the motion sequence, a suitable segmentation window is selected, and the segmentation points of the limb bone angle feature sequence are calculated according to the ARMA model. In this work, the limb bone angle information sequence of human skeletal posture is extracted, and the median filtering method is used to obtain the final set of segmentation points.

$$\begin{aligned} S_i = [S_a, S_b, S_c, S_d] &= \begin{bmatrix} S_{a_1} & S_{b_1} & S_{c_1} & S_{d_1} \\ S_{a_2} & S_{b_2} & S_{c_2} & S_{d_2} \\ S_{a_3} & S_{b_3} & S_{c_3} & S_{d_3} \\ S_{a_4} & S_{b_4} & S_{c_4} & S_{d_4} \end{bmatrix}, \\ s &= median(S_i). \end{aligned} \tag{7}$$

After automatically segmenting a complex motion sequence containing multiple actions into multiple sub-action sequences using the above segmentation method, a K-means clustering algorithm is used to extract key frames from the sub-action sequences. The K-means algorithm is an unsupervised learning algorithm [36] that is characterised by its simplicity of implementation and good clustering effect. The algorithm has a wide range of applications, and secondly, the K-means algorithm can effectively categorise similar frames in the motion sequence to achieve the purpose of key-frame extraction.

The K-means algorithm works by dividing the dataset into k class clusters, with the motion frames in each class cluster being the closest to the centroid of that cluster. For a sequence of features of motion $U = [u_1, u_2, \ldots, u_n]$ as input to the model, let each of these samples be of the same dimension and have the set of class clusters $C = [c_1, c_2, \ldots, c_n]$. The

K-means algorithm can partition these n samples into k class clusters, where $1 < k < n$, such that the intra-class sum of squared errors E is minimized.

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2, \tag{8}$$

where μ_i is the mean vector of class cluster C_i, i.e., the centre of mass of the class cluster, and the calculation of μ_i is given by Equation (9).

$$\mu_i = \frac{\sum_{x \in C_i} X}{|C_i|}. \tag{9}$$

During the execution of the algorithm, k points are randomly selected as the initial clustering centres; then, for each point in the dataset, it is calculated which centroid it is the closest to. In research, the Euclidean distance is one of the most commonly used measures of spatial distance, and the method is universal and applicable in all three dimensions.

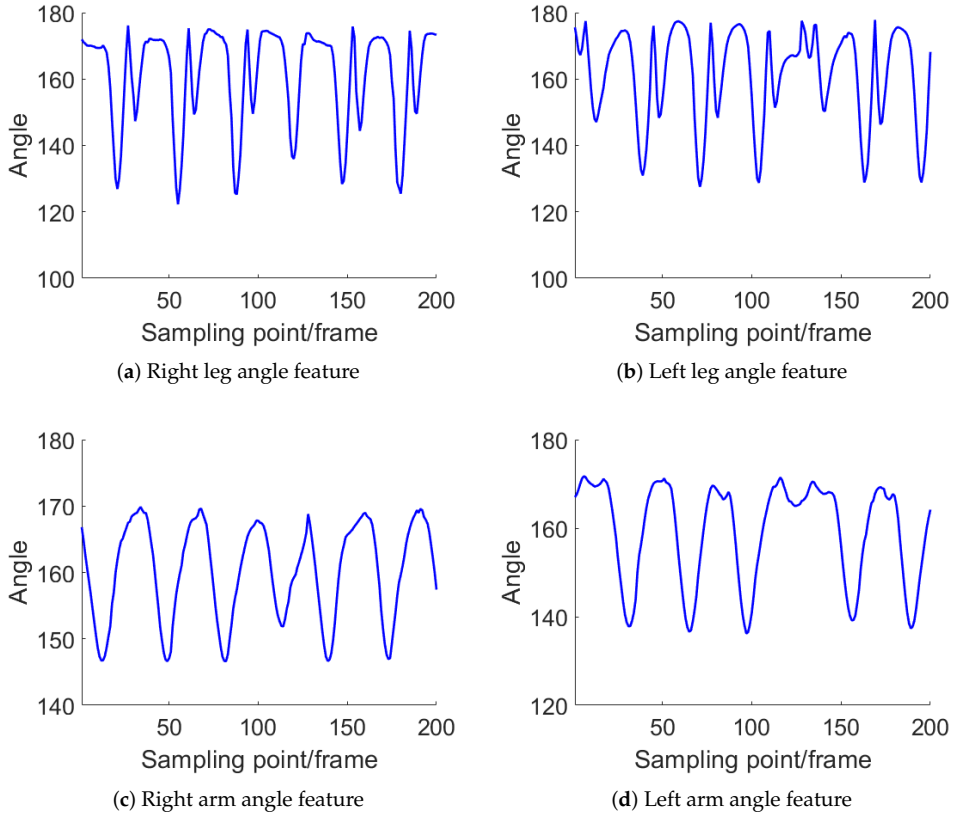
$$\mathrm{d}_{ij} = \left\|\mu_i - \mu_j\right\|_2^2 \tag{10}$$

where μ_i and μ_j are the mean vectors of class clusters C_i and C_j, respectively. All sample points in C are recalculated with a new centre of mass, μ_j, until all the centre-of-mass vectors no longer change and the final output is the reclassified class cluster $C = [\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_k]$. The determined k centre of mass is extracted as the key frame of this motion feature sequence.

3.2.2. Feature Coding Based on Motion Timing Features

In this chapter, the dimensionality of the motion feature matrix is reduced using label coding, and the key gestures of the extracted human actions are coded and assigned, so that the original sequence of feature vectors representing the action gestures is transformed into a sequence of digital codes, reducing the computational complexity to improve the rate of recognition of human actions. In the process of building the code table, we need to analyse the temporal characteristics of the movements in order to better encode them, as they have temporal characteristics. Taking walking as an example, Figure 2 shows the temporal angular characteristics of the human limbs during walking, from which it can be observed that the bone partitions of the same limb have a cyclical and causal nature in the temporal sequence. This can be used as a basis for determining the type of key gesture features.

For codebook building, a feature vector of key action gestures is first defined as F_{ak}, where a denotes the a action type and k denotes the k class of key gestures. The code table (codebook (CB)) contains the feature vectors of all action-critical poses, so the code table is also defined as $CB = \{F_{ik}\}, i = 1, \ldots, I$. The feature vectors of key action gestures in the code table are arranged according to the temporal order of the training sample data and include a total of k feature vectors. These key gestures are assigned as 1,2,…,K. The key gestures of different action types are converted into numerical sequences $\{c_1, c_2, \ldots, c_r\}$, thus achieving the purpose of encoding and allowing the human motion analysis method of this paper to be better generalised to various behavioural tasks.



(**a**) Right leg angle feature (**b**) Left leg angle feature

(**c**) Right arm angle feature (**d**) Left arm angle feature

Figure 2. Confusion matrix of recognition accuracy for each model.

3.3. LSTM Based on Key-Frame Attention Module

After the processing of human movement data using the modules mentioned above, we combined LSTM networks and introduced an attention mechanism to build a human dynamic action pose recognition model, named KF-LSTM (key-frame LSTM).

In this work, we used the LSTM network structure shown in Figure 3. A sequence of sample motion features of length N and dimension 20 was fed into the Bi-LSTM layer along with a sequence of labels of length N and dimension 1 containing 20 classes of action states. Feature sequences with 128 hidden states were obtained. The fully connected layer (FC) was then used to output a trained state matrix of length N and dimension 20. The output of the network was converted into a probability vector for each type of action state using a softmax layer. Finally, the relevant parameters of the different action types were obtained using a classification layer.

When training samples using LSTM networks [30], we have found that many action frames provide the same useful information on a large number of motion data and that some impressive action frames may contain the most discriminative information capable of recording the main action. We, therefore, used a key-frame-based attention mechanism for attention allocation and aggregated motion feature representations with different weights to reduce information fragmentation. Introducing attention into the human action model so that it gives more attention to key frames allows more effective human motion recognition to be achieved. Figure 4 shows a walking action sequence after extracting the key frames.

In the recognition task based on the KF-LSTM network, 11 representative human bone segments throughout the human body were selected in this paper to build the skeleton model of the human body, and the feature sequences were built according to this method. In this paper, the attention mechanism was applied to assign weights to the key frames generated under different types of motion. The attention mechanism is a weighted summation and a weighting mechanism that filters and extracts frames in the sequence that are more similar to the key frames and then reallocates the weights of these frames according to the

weighting values based on the attention mechanism. Specifically, the similarity of different frames to key frames in the feature sequence is used to determine their weighting in the reallocated weighting ratio.

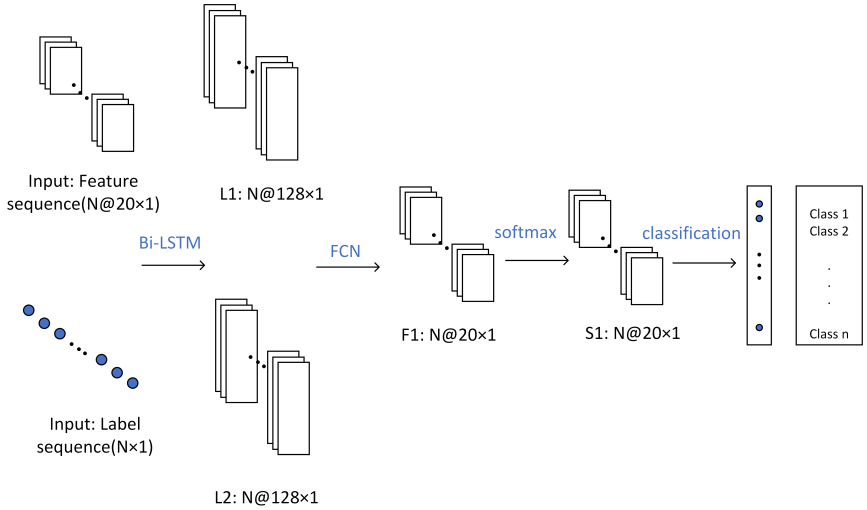


Figure 3. LSTM network architecture.

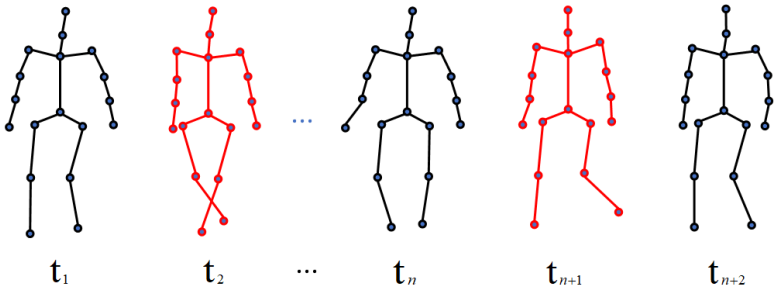


Figure 4. Note the walking action sequence after key-frame extraction.

The black skeleton image in Figure 4 is a normal frame in the walking sequence, and the red skeleton image is a key frame given a higher weight with the addition of the attention mechanism.

Assuming that the total number of frames of feature sequence U is n, the transformed feature sequence U' is obtained according to Equation (11) as

$$U' = [\alpha_1 U_1, \alpha_2 U_2, \ldots, \alpha_i U_i, \ldots \alpha_n U_n], \tag{11}$$

where U_i is the feature sequence before processing and α_i is the weight of each action frame. The key to the method is to calculate the appropriate α_i.

Q_i is the degree of correlation between each frame in the feature sequence and the key frame. When the degree of relevance is higher, the corresponding frame is assigned a higher weight value; when the degree of relevance is lower, the corresponding frame is assigned a lower weight value. The relevant calculation is given in Equation (12).

$$Q_i = \frac{cov(U_i, L_j)}{\sigma_U \sigma_L}, \; i \in [1, n]; j \in [1, 4], \tag{12}$$

where $cov(\bullet, \bullet)$ is the covariance between the normal frames and the key frames in the feature sequence, σ_U is the standard deviation of the normal frames in the feature sequence, and σ_L is the standard deviation of the key frames. After obtaining correlation degree Q_i

between each frame in the feature sequence and the key frame, the weight value is assigned to each frame in the final step.

$$\alpha_i = f(Q_i, U_i), \tag{13}$$

where $f(\bullet)$ is the weight assignment function between frames based on correlation Q_i and α denotes the output probability of the current sequence. It represents the final state of the action in the motion sequence and is used as the attention weight value for each action frame.

After assigning different weight scales to the action frames in the motion sequence, the action sequences with the noted weight values and the corresponding label numbers are fed into the LSTM network for training; the action frames with larger weight values receive a larger proportion of network training, and the action frames carrying more useful information are more likely to be output as recognition results. This greatly improves data utilisation and the performance of the recognition model.

4. Experimental Results and Analysis

4.1. Experimental Platform and Data Acquisition

This study used the Matlab-based data analysis programming language for network construction and algorithm design, and Axis Neuron Pro motion capture software (V2.10) for BVH data file exporting. A Perception Neuron Pro inertial motion capture device by Noitom was also used for data acquisition.

In order to design and evaluate the proposed action recognition system, MoCap data were measured on four subjects, including three males and one female. MoCap data [15] were collected with the Perception Neuron Pro model IMU MoCap device by Noitom Co. (Beijing, China). This device contained 17 IMUs and was located at the reference locations in Figure 5 [37]. Each IMU included internal adaptive filters and was calibrated before each measurement. Then, for motion segmentation analysis, the measurements were considered to contain negligible noise and bias effects. The sampling frequency of the measurements was configured to 100 Hz to cover the bandwidth of the body's major joint movements. Figure 6 shows the different types of action postures measured, including walking, running, hand raising, squatting, and leg lifting. In this work, the MoCap data we used referred to the human body as the experimental subject. The work in this paper is non-medically related research work; all the sensors were simply mounted on the body surface, and the human pose was recorded with the work of the sensor algorithm.

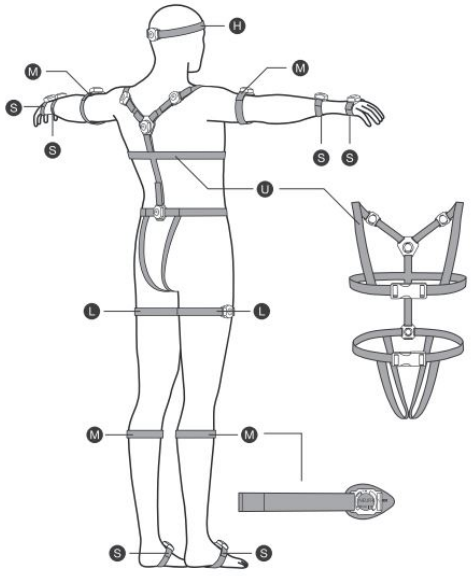
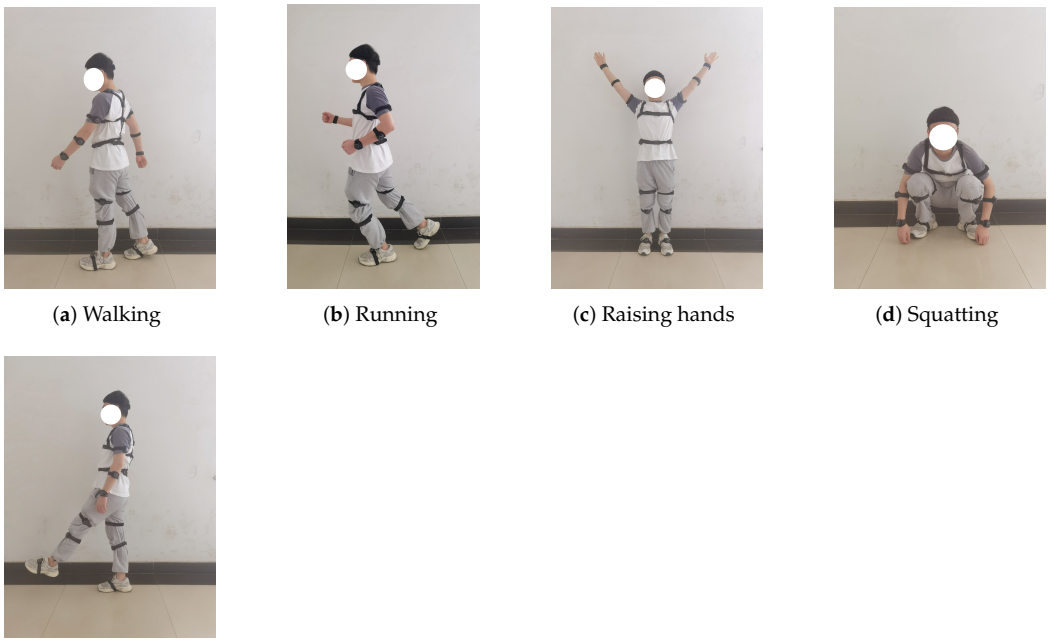


Figure 5. Diagram of the sensor wearing positions.



(**a**) Walking (**b**) Running (**c**) Raising hands (**d**) Squatting

(**e**) Leg raising

Figure 6. Different motion postures in motion sequences.

4.2. Analysis of Motion Sequence Segmentation Point Results

Figure 7 shows the waveforms of the angular features between adjacent bone segments in the human motion sequence in the time sequence. According to the variation in the angular magnitude, we can observe that this is a complex motion sequence containing multiple actions and that there are similar signal waveforms for different types of actions. There are often large errors in the localisation of action intervals, leading to the problem of confusion between frames in the time sequence on the subsequent clustering task, which has a serious impact on the clustering effect of the model. This has a serious impact on the clustering effect of the model. Therefore, we need to segment complex motion sequences before clustering and feature coding.

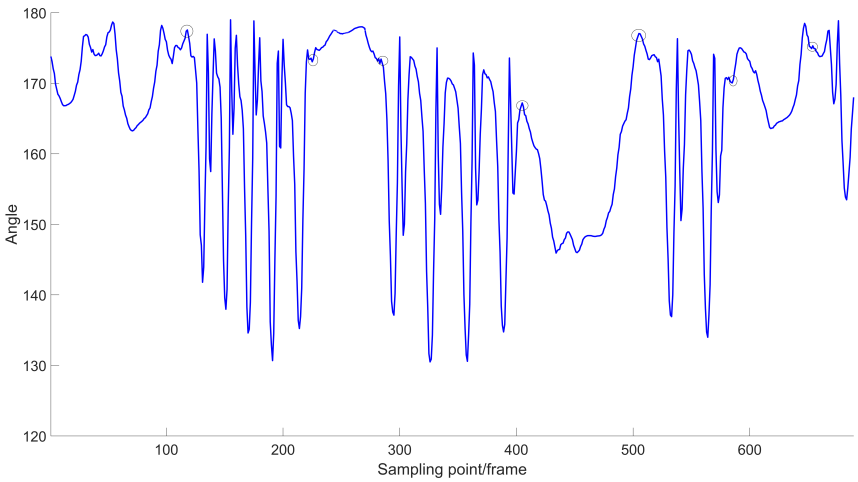


Figure 7. Characteristics of the angle between adjacent bone segments in the human motion sequence.

Figure 8 shows the segmentation points obtained after the calculation of the motion sequence performed by the ARMA-based automatic segmentation model, the segmentation of the motion sequence conducted by observing the presence of inflection points in the sequence, and finally the segmented segment-by-segment action sequence.

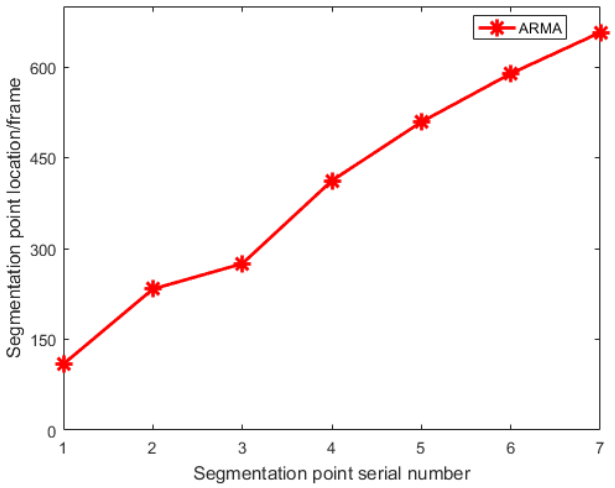


Figure 8. Determination of the segmentation points of the motion sequence.

4.3. Analysis of Clustering Results

Figure 9 shows the clustering results of the K-means algorithm on which this experiment was based. We used the pre-set parameters shown in Table 4 to cluster the motion sequences.

Table 4. K-means clustering parameter settings.

Parameter	Value
Input data dimension	20
Number of samples (frames)	234,000
Number of clusters	4
Action type	5

In this part of the work, we used the elbow method [38] to measure the cohesion of clusters and thus evaluate the clustering effectiveness of the K-means algorithm. The cohesiveness of a cluster is a key indicator of how closely related data are in the cluster. The core idea of the elbow method is that the larger the number of k class clusters is and the finer the sample division is, the more clustering each class gradually increases, and the progressively smaller the sum of squared errors naturally becomes. In the elbow method, the *WSS* derived from Equation (14) is used as an assessment indicator of the cohesiveness of class clusters.

$$WSS = \sum_{i} \sum_{x \in C_i} (x - \mu_i)^2, \tag{14}$$

The results of the elbow method on the five movements of walking, running, hand raising, squatting, and leg lifting are shown in Figure 10, where the y-axis labels are the *WSS* values and the x-axis labels are the values of the number of class clusters, k. It can be noted that there is a clear point of inflection for individual movements when the number of class clusters, k, is four. By continuing to increase the value of k after this point, there is no longer a significant change in intra-class error.

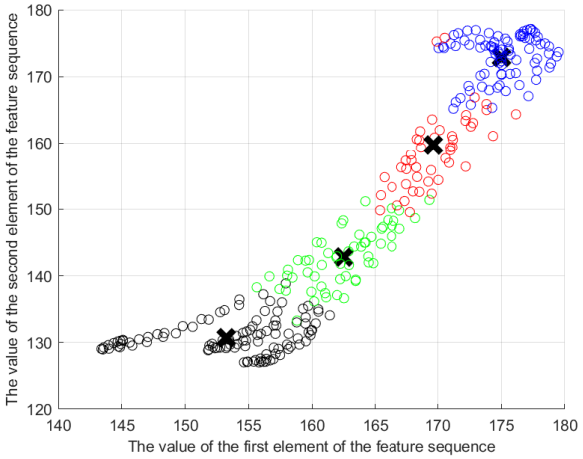
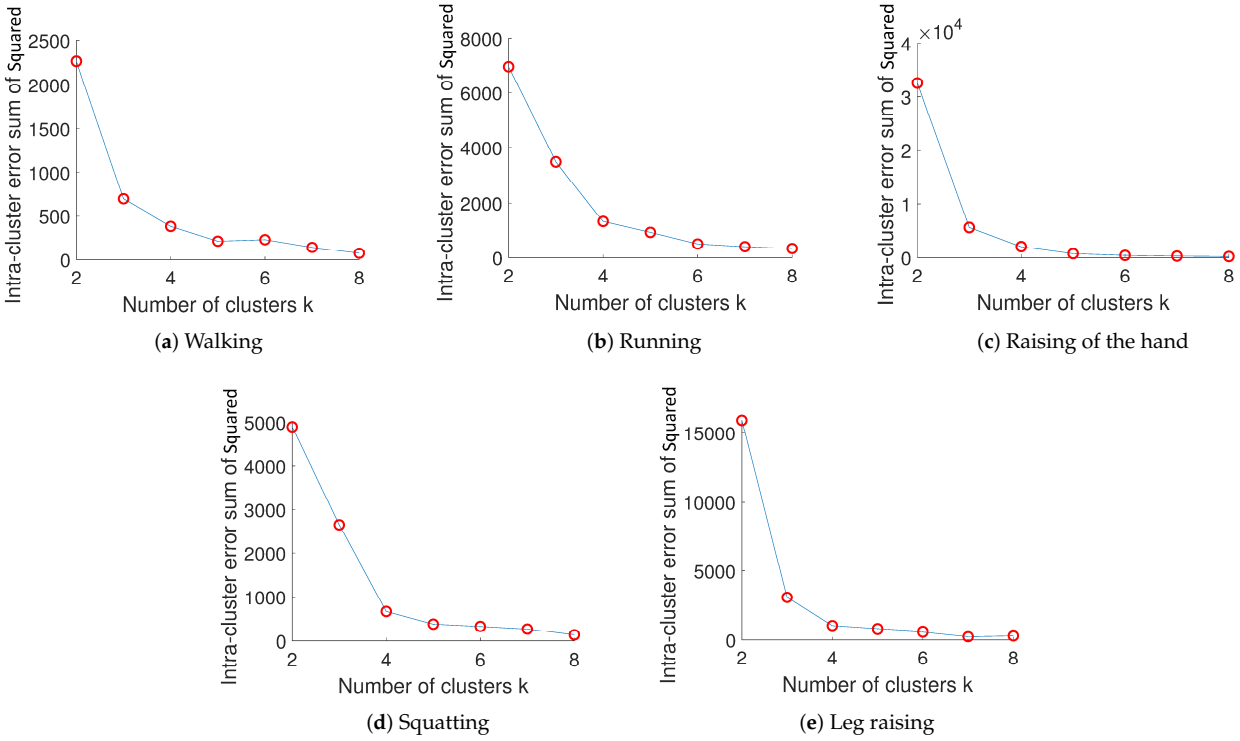


Figure 9. Distribution of clustering results for sample action sequences.



(**a**) Walking (**b**) Running (**c**) Raising of the hand

(**d**) Squatting (**e**) Leg raising

Figure 10. Results of the elbow method based on various types of movements.

4.4. KF-LSTM Parameter Setting

In general, the parameters of a network training model need to be set before training is considered. The selection of a reasonable set of parameters often enables the learning efficiency and training results of the network model to be optimized. The parameter settings for the network model in this paper are shown in Table 5 below.

Table 5. Training parameter setting.

Parameter	Value
Input size	20
Number of label categories	20
Max epochs	200
Input data number (frames)	234,000
Number of test sets	40

For training, we combined the processed motion data into a total training packet of approximately 234,000 frames, which was fed into the network for training. In order to improve the efficiency of data utilisation, a strategy of extracting 1 frame every 5 frames was used, resulting in a total training packet of approximately 46,700 frames. For the tests, the total number of test sets was 40, and the length of each test sequence was between 500 and 1600 frames. After the frame-drawing strategy, the length of each test sequence was between 100 and 400 frames.

4.5. Setting and Analysis of Correlation Recognition Models

To verify the recognition effectiveness of the proposed recognition method in this study on five action types, we compared it with an HMM-based approach [39], a 3D CNN-based approach [9], and a dual-stream attention-based LSTM approach [30] on human action recognition tasks for experimental and analytical discussion.

4.5.1. Identification Method Based on HMM Algorithm

In this study, the traditional HMM algorithm [39] was used in conjunction with a motor action database to perform the task of action recognition on test samples containing five types of actions: walking, running, hand raising, squatting, and leg lifting. For the human action recognition task, we used a left-to-right polymorphic Markov model. For a finite number of different pose states $S = \{S_1, S_2, \ldots S_N\}$, N is the number of states in the model, and the state at moment n can only be one of $\{S_1, S_2, \ldots S_N\}$. For a random vector $O = [O_1, O_2, \ldots O_T]$, where T represents the length of the time series, each observation vector has a corresponding output probability for a different state. Each action can be effectively modelled by a set of Hidden Markov Model parameters as $\lambda = (A, B, \pi)$. The Bayesian rule $P(O_i|\lambda)$ (where parameter A is a matrix representing the state transfer probabilities, parameter π is the initial state distribution probability, and parameter B represents the output probability of all states) is used as a way to calculate the probability of the actions generated by this model.

Let us suppose that the frequency count of the sample transfer from hidden state S_i to S_j is A_{ij}; then, the calculation of the state transfer matrix is given by Equation (15).

$$\begin{aligned} A &= [a_{ij}], \\ a_{ij} &= \frac{A_{ij}}{\sum_{s=1}^{N} A_{is}}. \end{aligned} \tag{15}$$

Let us suppose that the sample hidden state is S_j and the frequency count of observation state o_k is B_{jk}. We compare the similarity between each action frame in the observed state $O = [O_1, O_2, \ldots O_T]$ and the 20 types of key action frames extracted before using (14) to obtain the output probability of all states.

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}, \tag{16}$$

where m and n represent the rows and columns of each action frame, and $\bar{A}$ and $\bar{B}$ are the mean values of the state matrix and observation matrix, respectively.

Let us suppose that the frequency count of all samples with initial hidden state S_i is $C(i)$; then, the initial probability distribution is Equation (17).

$$\pi(i) = \frac{C(i)}{\sum_{S=1}^{N} C(s)}. \tag{17}$$

4.5.2. Recognition Method Based on 3D CNN

When using the 3D CNN approach [9] for recognition tasks, 3D convolution is achieved by convolving 3D kernels into a cube formed by superimposing multiple consecutive frames on top of each other. As shown in Figure 11, the input data dimension of each network layer is $N \times C \times T \times V \times M$, where N is the batch size of the data entering the network, C refers to the dimensional size of the features in each node, T refers to the maximum number of frames per sample sequence input, V refers to the number of nodes, and M refers to the number of coordinates. The input initial dimensions are $4 \times 1 \times 400 \times 18 \times 3$, and the features in spatial and temporal dimensions are extracted using four convolutional layers with convolutional kernel size (3, 3, 1) and convolutional kernel step (1, 1, 1), pooled, and then connected to a linear layer for classification, using cross-entropy as the loss function, finally obtaining a 4×5 category probability matrix; the maximum value of this dimension is taken as the final classification result.

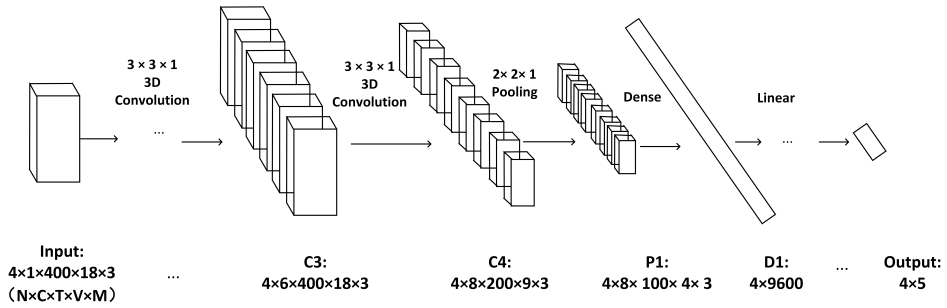


Figure 11. Three-dimensional CNN architecture for human action recognition.

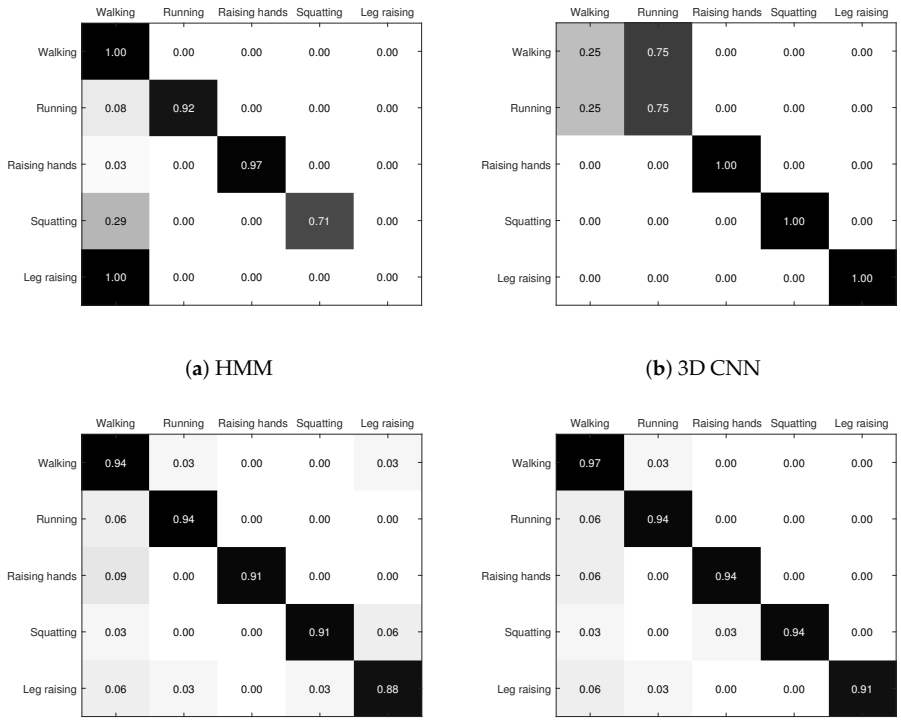
4.6. Experimental Results and Analysis of the Recognition Tasks

This experiment separately tested each recognition method using the test set, and the results are shown in the confusion matrix plot in Figure 12.

Figure 12 shows the confusion matrix for the recognition accuracy of the HMM-based method, the 3D CNN-based method, the dual-stream attention-based LSTM method, and our method on five types of action: walking, running, hand raising, squatting, and kicking. The numbers on the diagonal are the accuracy rates for each type of action. We take the average of the numbers on the diagonal of each confusion matrix as the average recognition accuracy of each recognition model. Table 6 demonstrates average recognition accuracy rates of 72.0% for the HMM-based method, 80.0% for the 3D CNN-based method, 91.6% for the dual-stream attention-based LSTM, and 94.0% for our method. It can be found that the recognition system constructed in this paper has high accuracy in the recognition task. Compared with past LSTM-based methods, it has gained some performance improvement and achieved the expected recognition results.

The HMM-based method has good results in the recognition tasks of walking, running, and hand raising. However, the recognition of the squatting and leg-lifting movements is too poor. This is due to the fact that HMM is an algorithm that relies entirely on the statistical characteristics of the data, and the parameters need to be set in the context of an existing action database, which may have a large bias towards the real environment. Moreover, the calculation of the probability distribution matrix relies on the judgement of similarity. In particular, for the leg-lifting action, there is an excessive correlation with the walking action. This also leads to the fact that purely statistical features cannot correctly determine the action category as a motion sequence of leg lifting. Regarding the 3D CNN-based approach, it has good recognition results in the recognition tasks of the action categories of hand raising, squatting, and kicking. However, in the recognition of the two movements of walking and running, there is a problem of mutual interference. This is also due to the high similarity between the two types of movement, and the model misidentified some of the test datasets as walking or running movements. The dual-stream attention-based LSTM approach worked well for the recognition of each action type.

However, relatively speaking, the method is slightly less effective than our method in the recognition of leg-lifting movements. Regarding our proposed recognition method, it can be found that the method has the highest recognition accuracy in the walking action type and has better recognition performance than other models in other actions. By comparing the overall recognition accuracy, it can be found that our method outperforms several other recognition methods in all recognition tasks, better recognises human action posture features in different motion states, and improves recognition accuracy.



(**a**) HMM (**b**) 3D CNN

(**c**) dual-stream attention-based LSTM (**d**) our method

Figure 12. Confusion matrix for recognition accuracy of different recognition methods.

Table 6. Comparison of average recognition accuracy.

Method	Average
HMM	72.0%
3D CNN	80.0%
LSTM	91.6%
KF-LSTM	94.0%

5. Conclusions

Performance improvement has always been a concern in human action recognition research. For our proposed recognition method, we first designed a key-frame extraction method based on the automatic segmentation model and the K-means clustering algorithm, which can accurately extract key frames of different actions and avoid inter-frame confusion between different key action frames and provides a reliable guarantee for the accuracy and robustness of the subsequent action recognition task of the KF-LSTM network. At the same time, we further optimised the network structure by combining the key-frame-based attention mechanism with the previous LSTM-based recognition method to construct a key-frame-based attention LSTM network. In the experimental section, we analyse and discuss the calculation of segmentation points, the clustering effect, and the comparison of the recognition accuracy of different recognition models. In this section, the calculation

process and results of the segmentation points are analysed; the variation in model performance and the rationality of K-means clustering with different K values are verified; finally, the recognition accuracy of our method is compared with that of several other methods using a test dataset containing five different motion states. The experimental results demonstrate that our method outperforms similar recognition models in terms of recognition performance and has better results in human motion recognition.

In the present work, only five different types of movement were used for the task of action recognition for experimentation and analysis, and in future work, additional types of movement will be considered to further improve research on relevant methods.

Author Contributions: Conceptualization, C.Y., F.M. and L.L.; methodology, C.Y. and F.M.; software, C.Y.; validation, C.Y., L.L. and T.Z.; formal analysis, C.Y. and L.L.; investigation, C.Y.; resources, C.Y., F.M. and L.L.; data curation, C.Y.; writing—original draft preparation, C.Y.; writing—review and editing, C.Y., L.L. and T.Z.; visualization, C.Y. and F.M.; supervision, L.L., J.T., N.J. and T.Z.; project administration, L.L.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Major Discipline Academic and Technical Leaders Training Program of Jiangxi Province (grant No. 20225BCJ22012), National Nature Science Foundation of China (grant No. 61801180), and Jiangxi Provincial Nature Science Foundation (grant No. 20202BAB202003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: All measurement data in this paper are listed in the content of the article, which can be used by all peers for related research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Antonik, P.; Marsal, N.; Brunner, D.; Rontani, D. Human action recognition with a large-scale brain-inspired photonic computer. *Nat. Mach. Intell.* **2019**, *1*, 530–537. [CrossRef]
2. Kwon, Y.; Kang, K.; Bae, C. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Syst. Appl.* **2014**, *41*, 6067–6074. [CrossRef]
3. Wang, P.; Liu, H.; Wang, L.; Gao, R.X. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. *CIRP Ann.* **2018**, *67*, 17–20. [CrossRef]
4. Barnachon, M.; Bouakaz, S.; Boufama, B.; Guillou, E. Ongoing human action recognition with motion capture. *Pattern Recognit.* **2014**, *47*, 238–247. [CrossRef]
5. Xia, L.; Chen, C.C.; Aggarwal, J.K. View invariant human action recognition using histograms of 3d joints. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 20–27.
6. Mozafari, K.; Moghadam Charkari, N.; Shayegh Boroujeni, H.; Behrouzifar, M. A novel fuzzy hmm approach for human action recognition in video. In Proceedings of the Knowledge Technology Week, Kajang, Malaysia, 18–22 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 184–193.
7. Li, X.; Zhang, Y.; Liao, D. Mining key skeleton poses with latent svm for action recognition. *Appl. Comput. Intell. Soft Comput.* **2017**, *2017*, 5861435 . [CrossRef]
8. Kansizoglou, I.; Bampis, L.; Gasteratos, A. Deep feature space: A geometrical perspective. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6823–6838. [CrossRef]
9. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef]
10. Tang, P.; Wang, H.; Kwong, S. Deep sequential fusion LSTM network for image description. *Neurocomputing* **2018**, *312*, 154–164. [CrossRef]
11. Liu, L.; Shao, L.; Rockett, P. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognit.* **2013**, *46*, 1810–1818. [CrossRef]
12. Jiang, M.; Pan, N.; Kong, J. Spatial-temporal saliency action mask attention network for action recognition. *J. Vis. Commun. Image Represent.* **2020**, *71*, 102846. [CrossRef]
13. Li, Q.; Lin, W.; Li, J. Human activity recognition using dynamic representation and matching of skeleton feature sequences from RGB-D images. *Signal Process. Image Commun.* **2018**, *68*, 265–272. [CrossRef]

14. Zhu, G.; Zhang, L.; Shen, P.; Song, J. Human action recognition using multi-layer codebooks of key poses and atomic motions. *Signal Process. Image Commun.* **2016**, *42*, 19–30. [CrossRef]
15. Mei, F.; Hu, Q.; Yang, C.; Liu, L. ARMA-Based Segmentation of Human Limb Motion Sequences. *Sensors* **2021**, *21*, 5577. [CrossRef]
16. Cheng, Y.B.; Chen, X.; Chen, J.; Wei, P.; Zhang, D.; Lin, L. Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
17. Ramasamy Ramamurthy, S.; Roy, N. Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1254. [CrossRef]
18. Wang, Y.; Sun, S.; Ding, X. A self-adaptive weighted affinity propagation clustering for key frames extraction on human action recognition. *J. Vis. Commun. Image Represent.* **2015**, *33*, 193–202. [CrossRef]
19. Gharahbagh, A.A.; Hajihashemi, V.; Ferreira, M.C.; Machado, J.J.; Tavares, J.M.R. Best Frame Selection to Enhance Training Step Efficiency in Video-Based Human Action Recognition. *Appl. Sci.* **2022**, *12*, 1830. [CrossRef]
20. Cho, T.Z.W.; Win, M.T.; Win, A. Human Action Recognition System based on Skeleton Data. In Proceedings of the 2018 IEEE International Conference on Agents (ICA), Salt Lake City, UT, USA, 18–22 June 2018; pp. 93–98.
21. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [CrossRef]
22. Zhang, H.B.; Zhang, Y.X.; Zhong, B.; Lei, Q.; Yang, L.; Du, J.X.; Chen, D.S. A comprehensive survey of vision-based human action recognition methods. *Sensors* **2019**, *19*, 1005. [CrossRef]
23. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4489–4497.
24. Zhu, Y.; Lan, Z.; Newsam, S.; Hauptmann, A. Hidden two-stream convolutional networks for action recognition. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 363–378.
25. Sarabu, A.; Santra, A.K. Distinct two-stream convolutional networks for human action recognition in videos using segment-based temporal modeling. *Data* **2020**, *5*, 104. [CrossRef]
26. Hu, H.; Cheng, K.; Li, Z.; Chen, J.; Hu, H. Workflow recognition with structured two-stream convolutional networks. *Pattern Recognit. Lett.* **2020**, *130*, 267–274. [CrossRef]
27. Meng, B.; Liu, X.; Wang, X. Human action recognition based on quaternion spatial-temporal convolutional neural network and LSTM in RGB videos. *Multimed. Tools Appl.* **2018**, *77*, 26901–26918. [CrossRef]
28. Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
29. Wang, X.; Miao, Z.; Zhang, R.; Hao, S. I3d-lstm: A new model for human action recognition. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Kazimierz Dolny, Poland, 21–23 November 2019; IOP Publishing: Bristol, UK, 2019; Volume 569, p. 032035.
30. Dai, C.; Liu, X.; Lai, J. Human action recognition using two-stream attention based LSTM networks. *Appl. Soft Comput.* **2020**, *86*, 105820. [CrossRef]
31. Oikonomou, K.M.; Kansizoglou, I.; Manaveli, P.; Grekidis, A.; Menychtas, D.; Aggelousis, N.; Sirakoulis, G.C.; Gasteratos, A. Joint-Aware Action Recognition for Ambient Assisted Living. In Proceedings of the 2022 IEEE International Conference on Imaging Systems and Techniques (IST), Kaohsiung, Taiwan, 21–23 June 2022; pp. 1–6.
32. Shah, A.; Mishra, S.; Bansal, A.; Chen, J.C.; Chellappa, R.; Shrivastava, A. Pose and joint-aware action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 3850–3860.
33. Li, J.; Liu, X.; Zhang, W.; Zhang, M.; Song, J.; Sebe, N. Spatio-temporal attention networks for action recognition and detection. *IEEE Trans. Multimed.* **2020**, *22*, 2990–3001. [CrossRef]
34. Muhammad, K.; Ullah, A.; Imran, A.S.; Sajjad, M.; Kiran, M.S.; Sannino, G.; de Albuquerque, V.H.C. Human action recognition using attention based LSTM network with dilated CNN features. *Future Gener. Comput. Syst.* **2021**, *125*, 820–830. [CrossRef]
35. Yasin, H.; Hussain, M.; Weber, A. Keys for action: An efficient keyframe-based approach for 3D action recognition using a deep neural network. *Sensors* **2020**, *20*, 2226. [CrossRef]
36. Sinaga, K.P.; Yang, M.S. Unsupervised K-means clustering algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
37. Axis Neuron User Guide. Available online: https://support.neuronmocap.com/hc/en-us/articles/10037078429595-Axis-Neuron-User-Guide (accessed on 6 June 2023).

38. Saputra, D.M.; Saputra, D.; Oswari, L.D. Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method. In Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Palembang, Indonesia, 16 November 2019; Atlantis Press: Amsterdam, The Netherlands, 2020; pp. 341–346.
39. Li, N.; Xu, D. Action recognition using weighted three-state Hidden Markov Model. In Proceedings of the 2008 9th International Conference on Signal Processing, Beijing, China, 26–29 October 2008; pp. 1428–1431.