SIN-CHN
SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2025, 22(1), 670.
https://doi.org/10.62617/mcb670

Article

The use of deep learning in intelligent athlete motion recognition: Integrating biological mechanisms

Ming Zhang, Yanfeng Li, Yuhong Cui[*]

Department of Sports Training, Hebei sport university, Shijiazhuang 050041, China
*** Corresponding author:** Yuhong Cui, yuhongcui_hepec@126.com

CITATION

Zhang M, Li Y, Cui Y. The use of deep learning in intelligent athlete motion recognition: Integrating biological mechanisms. Molecular & Cellular Biomechanics. 2025; 22(1): 670. https://doi.org/10.62617/mcb670

ARTICLE INFO

Received: 30 October 2024
Accepted: 15 September 2024
Available online: 6 January 2025

Abstract: This work explores the effective application of deep learning for recognizing athletes' movements, aiming to enhance precision in competitive sports. Traditional motion analysis methods primarily rely on manual observation, which can introduce subjective bias and limit accuracy. To address these limitations, we propose an automated method based on deep learning for recognizing and classifying athletes' technical movements while evaluating their performance. A hybrid model, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, is utilized to extract key frames from video data. The CNN is responsible for feature extraction, capturing the intricate details of movement, while the LSTM captures the temporal sequence characteristics, providing context to the actions. To further strengthen our approach, we delve into the biological mechanisms underlying athletic movements. Understanding the biomechanics of motion—such as joint angles, muscle activation patterns, and energy expenditure—can enhance the accuracy of deep learning models. By integrating these biological insights into our model, we improve the recognition process, allowing for a more nuanced understanding of how movements impact performance. Through experiments, we demonstrate that the model achieves high accuracy across multiple benchmark datasets (UCF-101, HMDB-51, Kinetics-400, and Sports-1M), with a particularly high accuracy of 93.5% on the UCF-101 dataset. These results indicate that the proposed method is both accurate and reliable, making it suitable for athlete training and competition analysis. The findings of this research have significant implications for sports science, training evaluation, and injury prevention. By providing coaches and athletes with precise feedback based on deep learning analysis, we can facilitate targeted training interventions that enhance performance while reducing injury risks. This work aims to offer a powerful tool for athletes, coaches, and researchers, contributing to the advancement of competitive sports through a deeper understanding of movement dynamics and their biological underpinnings.

Keywords: deep learning; athlete skill analysis; biomechanics; computer vision; motion recognition; injury prevention

1. Introduction

Precise motion analysis is critical for enhancing athlete performance in competitive sports. Advancements in technology, particularly breakthroughs in computer vision and artificial intelligence, have ushered humanity into a new era where athlete performance can be assessed more accurately and scientifically through automated tools [1–3]. Traditional motion analysis methods typically rely on manual observation and annotation by experts. This is not only time-consuming and labor-intensive but also subject to the observer's subjective judgment and personal experience, making it difficult to ensure consistency. Moreover, traditional methods are often limited to specific conditions, such as laboratory environments, where the results may not fully reflect an athlete’s performance in real competition settings [4–6]. Recently, as Deep Learning (DL) technology has advanced, it has opened up new possibilities for using Machine Learning (ML) for the automatic analysis of athletic skills. DL, an ML method based on artificial neural networks, can learn complex patterns and feature representations from large amounts of unlabeled data, making it highly suitable for processing image and video data [7,8]. In the field of competitive sports, DL has been successfully applied to various aspects such as action recognition, pose estimation, and motion tracking, providing strong technical support for athlete training. Additionally, DL technology helps address two major issues in traditional methods: Data collection convenience and the objectivity of analysis results. Portable devices and smart wearable technology make it easy to collect athletes’ training and competition data. Meanwhile, DL models can provide consistent and reliable analysis results without human interference, which is crucial for improving training efficiency and competition outcomes [9–11].

This work aims to explore how DL technology can be utilized for efficient athlete skill and motion analysis. Developing a DL-based motion recognition system enables real-time monitoring and evaluation of athlete movements. The foundation of this system is rooted in using the Convolutional Neural Network (CNN) to extract motion features from video data and applying the Recurrent Neural Network (RNN) to capture the temporal characteristics of the movements. This combination not only enables the recognition of individual actions but also understands transitions between actions, providing coaches with a more comprehensive motion analysis report.

However, despite the advantages of DL, it also faces some challenges. The first challenge is the need for a large amount of data, as DL models usually require extensive training data to achieve good performance. The second challenge is the issue of model interpretability. DL models are often regarded as “black boxes”, meaning that even if the model makes correct predictions, it is difficult to understand how it arrived at the conclusion. The third challenge is the demand for computational resources, as training large-scale DL models requires high-performance computing equipment, which could be a significant obstacle for smaller sports clubs.

Therefore, this work also explores how to address these challenges by proposing an innovative method to balance performance and resource consumption. The work discusses how to use transfer learning to decrease the amount of data needed for training, design models with strong interpretability to better understand model behavior, and optimize model structures to suit different hardware platforms. To enhance the transparency and interpretability of the model, feature visualization and attention mechanisms are incorporated to help understand the decision-making process during motion recognition. In the feature extraction phase, the Gradient-weighted Class Activation Mapping (Grad-CAM) method is used to visualize the feature maps generated by the CNN. By analyzing these feature maps, it becomes possible to identify the specific regions that the model focuses on during key frames. For example, when recognizing a basketball player’s shooting motion, Grad-CAM shows the model’s attention distribution on the hands and the ball, helping to clarify the reasoning behind its decisions. In the LSTM network, an attention mechanism is introduced to enable the model to better focus on the features at key moments when processing action sequences. Specifically, attention weights are calculated for each time step to determine the impact of each key frame on the current state. This allows the model to emphasize critical moments in an action, such as the jump, shot release, and landing during a basketball shooting motion. This approach not only improves the model's performance but also enhances its interpretability, enabling coaches and athletes to understand the basis for the model's judgments. Through these efforts, this work is expected to foster the broad adoption of DL technology in sports, providing more powerful tools for athletes, coaches, and researchers, and contributing to the development of competitive sports.

2. Literature review

2.1. The use of ML and DL in athlete motion analysis

In recent years, ML and DL technologies have been extensively utilized in the field of athlete motion analysis. Traditional ML methods, like the Support Vector Machine (SVM), Decision Tree, and Support Vector Regression, have achieved initial success in motion recognition and classification. Xu utilized SVM to classify swimming strokes and found that this method achieved high accuracy in distinguishing different strokes [12]. However, these traditional methods often require manual feature engineering, which limits their generalization ability and robustness. In contrast, DL methods, with their powerful feature learning capabilities, can directly extract useful features from raw data, thus demonstrating superior performance.

DL technologies, particularly the CNN and RNN, have shown immense potential in motion recognition and behavior analysis. CNN excels at handling image and video data and automatically learning local features within images, while RNN is proficient in processing time-series data, and capturing the dynamic changes in movements. Ullah and Munir proposed a dual-stream CNN architecture that combines spatial and temporal streams to analyze motion videos, significantly improving the accuracy of action recognition [13]. Additionally, Zan and Zhao used LSTM networks to model motion videos, effectively recognizing complex action sequences by learning the temporal dependencies of movements [14]. In addition to traditional CNN and LSTM architectures, Lovanshi and Tiwari explored the application of the Graph Convolutional Network (GCN) in motion recognition [15]. Their findings showed that GCN performed exceptionally well in capturing the relationships between human body keypoints during motion, enabling more accurate recognition of various sports actions. Furthermore, Xin et al. proposed an ensemble learning approach that combined different DL models to improve the robustness of action recognition [16]. Their research demonstrated that integrating multiple models effectively reduced the limitations of a single model and enhanced overall recognition accuracy.

2.2. Advantages and limitations of different methods

While DL methods have demonstrated outstanding performance in motion analysis, they also come with their own set of advantages and limitations. Ahmed et al. suggested that one of the notable advantages of DL methods was their powerful feature learning capability, which allowed them to automatically learn high-level abstract features from data. This gives them a natural advantage when dealing with unstructured data [17]. Taye highlighted that the flexibility of DL models allowed them to adapt to a wide range of application scenarios. Whether dealing with static images or dynamic videos, these models can be tailored to handle various tasks through appropriate network architectures [18].

In terms of recognizing the contributions of female researchers, Carvalho et al. proposed a DL model for swimming, which revealed that there were differences in the technical movements between female and male swimmers [19]. This finding provided important insights for customizing personalized training programs in the future. In addition, Legault and Faubert conducted a study on athletics, highlighting the impact of gender on sports movement recognition, and offering valuable references for improving the technical performance of female athletes [20]. From a regional perspective, Lee et al. implemented DL technology in sports training to explore its role in improving athletic performance [21]. The results indicated that DL models not only improved the accuracy of action recognition but also provided coaches with targeted training feedback, demonstrating the practical applicability and operability of the technology.

However, DL methods also exhibit some significant limitations. These models generally demand substantial amounts of data for training, and acquiring high-quality labeled data is frequently a time-consuming and expensive process [22]. Additionally, DL models are often considered “black box” models, lacking transparency, which makes their decision-making process difficult to interpret. This is particularly important in the field of sports, where coaches and athletes need to understand how the model arrives at its assessments [23]. Moreover, DL models tend to perform poorly in situations with small sample sizes. When training data are insufficient, the models are prone to overfitting. On the other hand, traditional ML methods, although more dependent on manual feature engineering, generally offer better interpretability, making them easier to understand and debug. For certain specific tasks, such as simple action classification, traditional ML methods may be sufficient and are more suitable in scenarios where computational resources are limited.

2.3. Current challenges and future research directions

The field of athlete action analysis faces several key challenges, including data acquisition, model interpretability, and the demand for computational resources. First and foremost, efficiently obtaining high-quality training data remains a significant obstacle. Although advancements in sensor and video recording technologies have made data collection more convenient, data annotation still requires substantial human effort. Future research could explore automatic annotation techniques to reduce the need for manual involvement. Another critical challenge is model interpretability. Many current DL models operate as black boxes, which poses a bottleneck for practical applications. Future research should pay attention to exploring new methods to enhance model transparency, making the decision-making process of these models more understandable. For instance, integrating DL with rule-based learning could be a potential approach to explain model behavior. The demand for computational resources is also a pressing issue. While DL models deliver exceptional performance, they often require high-performance computing devices for training and deployment. A future trend may involve developing lightweight model architectures that can run on embedded devices, thereby reducing hardware costs. Despite these challenges, the role of DL-based athlete action analysis in sports science research is expected to grow significantly in the future. Overcoming existing limitations and developing new methodologies can provide athletes with more precise and effective training guidance, ultimately advancing the level of competitive sports. Therefore, this work intends to explore an efficient and interpretable DL approach to achieve a precise analysis of athlete skills and actions, offering practical solutions for real-world applications.

3. Method

3.1. Data preparation and preprocessing

To ensure that the model can extract meaningful information from video data, the following steps are undertaken for data preparation and preprocessing:

1) Video Data Collection: Video data form the foundation of this work. This work collects a substantial number of video clips from various sports events and training sessions. To cover a broad range of sports scenarios, the video data include different sports such as football, basketball, athletics, and swimming. Each video clip contains a complete action cycle to guarantee the diversity and representativeness of the data.
2) Key Frame Extraction: Extracting key frames from videos is a crucial preprocessing step. Key frames are specific frames that represent the main content of a video segment. This work uses the frame difference method to determine key frames. Specifically, by calculating the differences between adjacent frames, frames with significant differences are identified as key frames. This method effectively removes redundant information and reduces the amount of data for subsequent processing.

It is assumed that the video frame sequence is denoted as $\{I_1, I_2,..., I_n\}$, where I_i represents the i-th frame image. The frame difference can be defined as:

$$D_i = \|I_{i+1} - I_i\|_2 \tag{1}$$

$\|.\|_2$ represents the Euclidean distance. If D_i is greater than a certain threshold T, I_i is considered a key frame. The threshold T is adjusted based on the actual video content to ensure that the extracted key frames are representative.

3) Data Annotation: With the purpose of training the DL model, the video data need to be annotated. The annotation process involves classifying and labeling the actions in each key frame. Multiple professional sports coaches and athletes are invited to participate in the annotation process to ensure accuracy and consistency. The annotation includes, but is not limited to, actions such as starting, sprinting, jumping, throwing, catching, and passing.
4) Data Augmentation: To improve the diversity and richness of the data, data augmentation techniques are adopted. These techniques involve rotation, translation, scaling, and flipping to simulate different perspectives and environmental conditions. Data augmentation generates additional training samples, improving the model’s robustness and generalization ability.

3.2. Feature extraction and model construction

After completing data preparation and preprocessing, the next stage involves feature extraction and model construction.

1) CNN Architecture: This work employs CNN as the foundational architecture for processing video data. CNN is a DL model particularly suited for handling image and video data, as it can automatically learn local features within images [24]. **Figure 1** illustrates the CNN architecture.

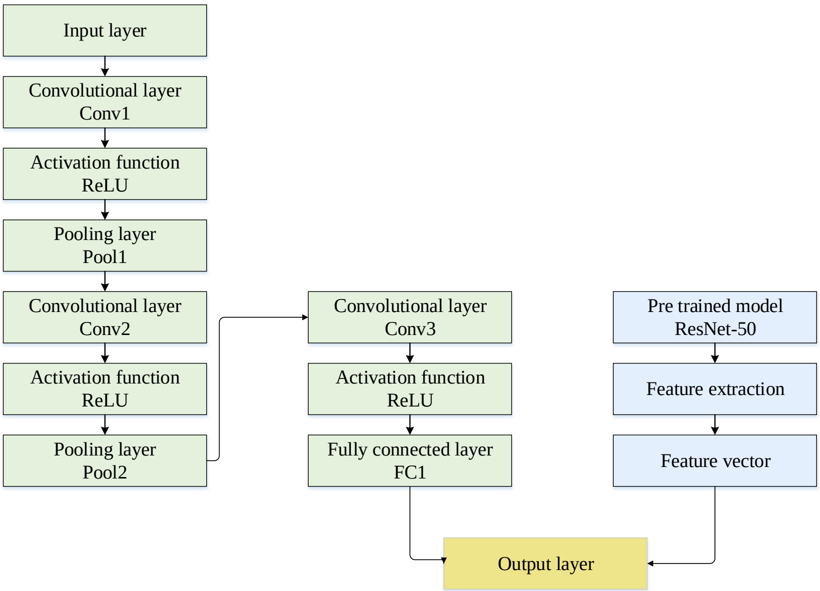


Figure 1. CNN architecture.

In CNN, convolutional layers use the concept of local receptive fields, meaning each neuron is connected to only a small portion of the input data. This local connectivity allows the model to focus on specific regions within an image and learn useful features from them. For example, when recognizing an athlete’s movements, the model can concentrate on details of the arms, legs, or other key body parts, capturing subtle differences in the action. The weights in convolutional layers are shared, meaning the same filter slides across the entire input image to detect similar types of features. This weight-sharing mechanism decreases the parameter quantity, making the model more efficient and capable of detecting translational invariance in images. For instance, when recognizing a runner’s motion, the model can identify the same features regardless of where the motion occurs in the image. CNN is typically composed of alternating convolutional layers and down-sampling layers, forming a hierarchical structure. Each layer learns different levels of features, from edges and textures to higher-level shapes and objects. This hierarchical feature learning enables CNN to handle complex image and video data. For example, when identifying the motion of a high jumper, the first layer might learn edge features of the legs and arms, while subsequent layers may learn the overall pose and dynamic changes of the action. To address the issues of vanishing and exploding gradients in deep networks, Deep Residual Network (ResNet) is used. ResNet introduces residual blocks that make it easier for the network to learn identity mappings, thus avoiding degradation in training deep networks. ResNet-50 is selected as the pre-trained model. It is a ResNet that has been pre-trained on the ImageNet dataset, offering strong feature extraction capabilities.

2) Application of Pre-trained Models: To fully leverage the advantages of the pre-trained model, ResNet-50 is first used to extract features from the key frames. Specifically, each key frame is fed into the ResNet-50 model, and the feature vector from the penultimate layer is extracted. These feature vectors contain rich visual information that can be used to describe the action characteristics within the key frames. x_i represents the i-th key frame, and after processing through ResNet-50, the extracted feature vector is denoted as $f(x\_i\ )$. It is assumed that $f(x_i) \in R^d$, where d is the dimension of the feature vector.
3) Feature Fusion and Sequence Modeling. Since actions are composed of a series of continuous key frames, it is essential to consider the temporal sequence characteristics of the actions. An RNN is employed to model the action sequence. Specifically, an LSTM network is used to capture the temporal dependencies of the actions. LSTM is a specialized type of RNN that effectively handles long-term dependencies [25]. LSTM addresses the issues of vanishing and exploding gradients that traditional RNNs face when processing long sequences by introducing input, forget, and output gate mechanisms. The basic unit of an LSTM consists of a cell state and three gating mechanisms that control the flow of information. The cell state preserves long-term dependencies within the sequence, while the input, forget, and output gates determine which information should be written, forgotten, and read, respectively. Specifically, the input gate decides which information should be written into the cell state, the forget gate determines which information should be removed from the cell state, and the output gate selects which information should be read from the cell state to generate the output for the current time step. **Figure 2** illustrates the LSTM architecture.

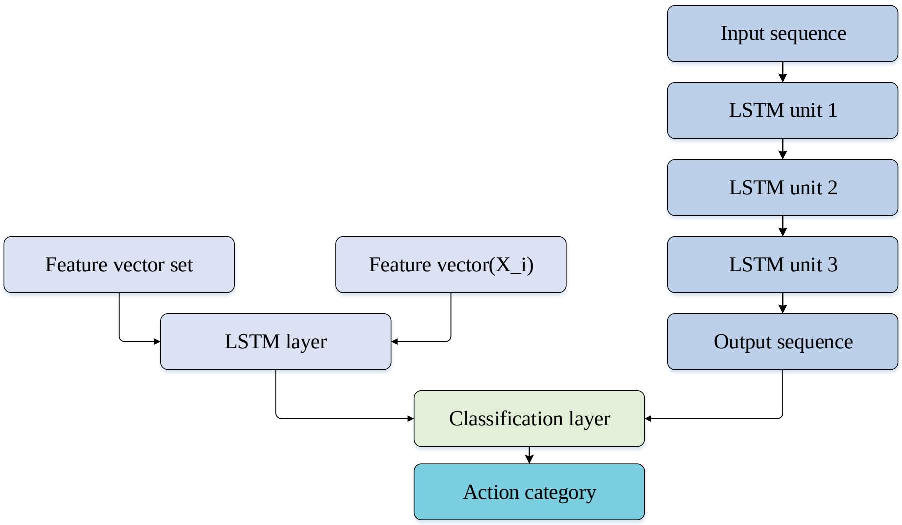


Figure 2. LSTM architecture.

$F = \{f(x_1), f(x_2),..., f(x_m)\}$ represents the set of feature vectors for an action sequence, where m is the number of key frames. The hidden state in the LSTM is updated according to the following equation:

$$h_t = LSTM(f(x_t), h_{t-1}) \tag{2}$$

h_t represents the hidden state at time t, h_{t-1} is the hidden state at time t−1, and $f(x_t)$ is the feature vector at time t. The internal mechanism of the LSTM includes the input gate, forget gate, and output gate, which are represented by the following equations:

$$i_t = \sigma(W_{xi} \cdot f(xt) + W_{hi} \cdot h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf} \cdot f(xt) + W_{hf} \cdot h_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo} \cdot f(xt) + W_{ho} \cdot h_{t-1} + b_o) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc} \cdot f(xt) + W_{hc} \cdot h_{t-1} + b_c) \tag{6}$$

$$h_t = o_t \odot tanh(c_t) \tag{7}$$

σ is the sigmoid function, $\odot$ represents element-wise multiplication, and W_{xi}, W_{hi}, and b_i are weight matrices and bias terms. This work leverages LSTM to capture the temporal characteristics of athletes' movements, thereby improving the understanding and recognition of complex action sequences. For instance, when recognizing a basketball player's shooting motion, LSTM can identify the key temporal features, such as the arm raising, the moment of the shot, and the arm lowering, which enhances the accuracy of action recognition. Additionally, LSTM can handle action sequences of varying lengths, providing the model with greater robustness and generalization ability. By using feature vectors extracted from a CNN as inputs to the LSTM, the model can comprehensively consider both the visual features and the temporal characteristics of the actions, enabling precise analysis of athletic skills and movements.

In model construction, a CNN-LSTM hybrid architecture is selected because this combination effectively captures both spatial and temporal features of motion data. The CNN is capable of extracting local features from each frame, while the LSTM leverages these features to capture the temporal dependencies within the action sequence, thereby enhancing the model's ability to recognize complex movements. ResNet-50 is chosen as the backbone network for the following reasons: ResNet-50 employs a residual learning mechanism, which effectively addresses the vanishing gradient problem in deep networks, ensuring that the network performs well even in deeper layers. Additionally, ResNet-50 has been pre-trained on the ImageNet dataset, making it a powerful feature extractor that aids in faster convergence and improved model accuracy. During training, the following key parameters are set: a learning rate decay strategy is used, with the initial learning rate set to 0.001 and halved every 10 epochs. This strategy helps fine-tune model parameters in the later stages of training, improving convergence precision. A batch size of 32 is chosen, providing a good balance between computational efficiency and memory consumption, while also aiding the model's generalization during training. The Adam optimizer is used, as its adaptive learning rate adjustment mechanism effectively adjusts the learning rate for each parameter during training, accelerating model convergence. These choices have a significant impact on the model's performance. Proper learning rate scheduling and optimization help improve training stability and speed, while an appropriate batch size contribute to better model generalization.

3.3. Model training and validation

After completing feature extraction and model construction, the subsequent step is to train and validate the model.

(1) Model Training: The extracted feature vectors and corresponding labels are fed into the LSTM model for training. To prevent overfitting, regularization techniques such as Batch Normalization and Dropout are employed. Batch normalization is a commonly used regularization technique in neural network training that accelerates the training process and enhances model stability by standardizing the activation values of each batch of data. Specifically, batch normalization standardizes the activation values of each layer to have zero mean and unit variance, reducing internal covariate shifts and making the model easier to converge. In the model, batch normalization is applied to both the input and hidden layers of the LSTM. For example, in the LSTM's input layer, the feature vectors extracted from the CNN are normalized to ensure the stability and consistency of the input data. Dropout is another widely used regularization technique that mitigates model overfitting by randomly dropping a portion of neurons during training. In this process, each neuron has a certain probability of being temporarily "dropped", meaning its output is set to zero. This random dropout mechanism forces the model to learn more robust feature representations, thereby enhancing its generalization ability. Here, dropout is applied to the hidden layers of the LSTM model. Specifically, in each training batch, a certain percentage of neurons (such as 50%) are randomly dropped to prevent the model from relying too heavily on certain specific feature representations. By combining batch normalization and Dropout, the model can maintain its performance while effectively preventing overfitting. Batch normalization is also applied between the hidden layers of the LSTM to further improve the model's stability and generalization ability. Additionally, Early Stopping [26] is utilized to monitor the training process, terminating it when the performance on the validation set no longer improves. It is a commonly used technique to prevent overfitting. The basic idea is to periodically evaluate the model's performance on the validation set during training and to stop the training process when the performance no longer improves. Specifically, during the training process, the model's performance is evaluated on the validation set at regular intervals (such as every 5 epochs). If the performance on the validation set (such as accuracy or loss value) does not show significant improvement over several consecutive epochs, it is considered that the model has begun to overfit, and the training process is terminated early. This approach allows the training to stop before the model starts overfitting, resulting in a model with stronger generalization ability. Here, Early Stopping criteria are set, such as terminating the training if the accuracy on the validation set does not improve for 10 consecutive epochs. The loss function L can be defined as cross-entropy loss:

$$L(y,\hat{y}) = -\sum_{i=1}^{N} y_i log(\overset{\wedge}{y_i}) \quad (8)$$

y represents the true labels, $\hat{y}$ denotes the model’s predicted probability distribution, and N is the number of classes.

(2) Cross-Validation: To assess the model’s generalization capability, cross-validation is conducted across multiple datasets. Specifically, k-fold Cross Validation is employed, where the dataset is divided into k subsets. In each training iteration, k-1 subsets are taken as the training set, with the remaining subset serving as the validation set. Through multiple training and validation rounds, the average performance metrics of the model across different datasets are obtained.

$S = \{S_1, S_2,..., S_k\}$ represent the k subsets, and the cross-validation process can be expressed as follows:

$$Performance = \frac{1}{k}\sum_{i=1}^{k} Evaluate(S_i, S\backslash Si) \quad (9)$$

$\sum_{i=1}^{k} Evaluate(S_i, S\backslash Si)$ represents the performance metric in the *i*-th validation round.

(3) Performance Evaluation. To assess the performance of the model, several performance metrics are used, like Accuracy, Precision, Recall, and F1 Score. Accuracy reflects the proportion of correctly classified instances; Precision measures how many of the samples predicted as positive are truly positive; Recall gauges how many of the actual positives the model correctly identifies; and F1 Score is the harmonic mean of Precision and Recall, providing a balanced measure of both metrics.

Experiments are conducted using four publicly available datasets. UCF-101: Contains action videos across 101 categories, covering a variety of sports activities. HMDB-51: Includes action videos in 51 categories, mainly used for evaluating action recognition performance. Kinetics-400: A large-scale video dataset with 400 action categories. Sports-1M: Comprises over 1 million sports video clips, covering a broad range of sports activities.

To validate the superiority of the established model, comparisons are made with several other methods, including Traditional Methods: Using SVM for action recognition. CNN: Utilizing a standard CNN for action recognition. Two-Stream CNN: The Two-Stream CNN architecture proposed by Simonyan and Zisserman. LSTM-Based Method: Employing LSTM networks for modeling action sequences.

4. Results

4.1. Overall performance on datasets

The overall effectiveness metrics of the established model on various datasets are analyzed. **Figure 3** shows the results.

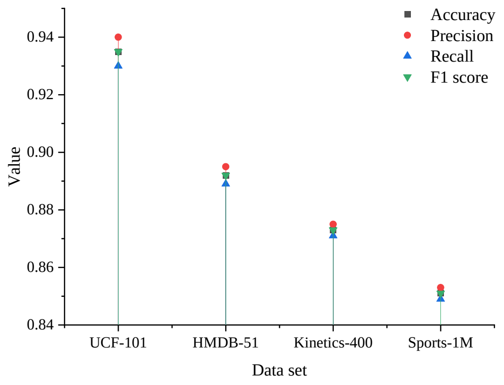


Figure 3. Overall performance on datasets

Figure 3 suggests that the proposed model performs exceptionally well across all datasets. Its performance varies significantly across different datasets. The UCF-101 dataset contains 101 action categories, covering a wide range of sports types. However, most of the actions are relatively simple and exhibit clear visual features, enabling the model to recognize these actions effectively, achieving a high accuracy rate of 93.5%. The HMDB-51 dataset, although containing fewer action categories, includes more complex movements such as turning and quick direction changes. The complexity of these actions leads to a slight decrease in model accuracy, which reaches 89.2%. The Kinetics-400 dataset includes 400 action categories, increasing the diversity and complexity of the actions. As a result, the model's performance is affected by some less common actions, and accuracy drops to 87.3%. Particularly for actions like jumping and throwing, the variety in how these actions are presented in videos causes the model to make more misclassifications.

4.2. Comparison of different models

The effectiveness of the proposed model is compared with several other methods across different datasets. **Figure 4** presents the results.

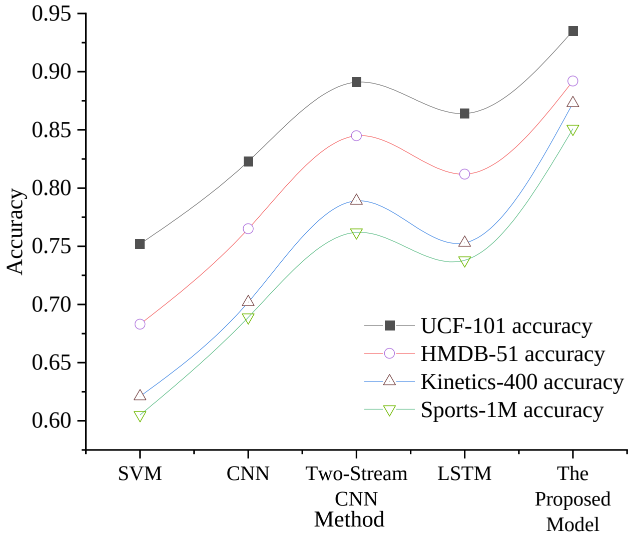


Figure 4. Comparison of different models.

Figure 4 shows that the proposed model achieves significantly higher accuracy across all datasets compared to other methods. For example, on the UCF-101 dataset, the proposed model's accuracy is 93.5%, whereas SVM and CNN achieve 75.2% and 82.3% respectively, and Two-Stream CNN and LSTM reach 89.1% and 86.4%. Similarly, on other datasets, the proposed model consistently demonstrates higher accuracy. This indicates that the proposed model has a notable advantage in action recognition tasks.

4.3. Error rate analysis of the model

The error rates for different action categories are analyzed. **Figure 5** presents the results.

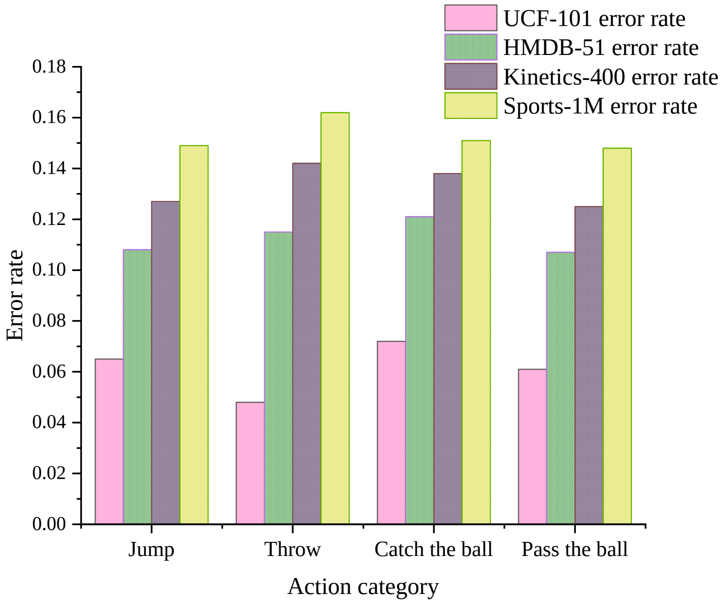


Figure 5. Model error rate analysis.

Figure 5 shows that when analyzing the error rates across different action categories, the error rate for jumping actions is significantly higher than for other categories. The Sports-1M dataset is large, but due to the diversity in video content and variations in sports styles, the model's accuracy on complex actions is relatively lower. Especially for jumping actions, the model's error rate is the highest, reaching 14.9%. This can be attributed to factors such as the uncertainty in the motion's changes during the jump, the influence of camera angles on action recognition, and variations in speed throughout the action. Therefore, it is necessary to increase the number and diversity of training samples for these complex actions to improve the model's accuracy.

4.4. Challenges in jumping action recognition

In the experiments, it is observed that the error rate for jumping actions is significantly higher than for other action types, particularly reaching 14.9% on the Sports-1M dataset. In response, an in-depth analysis is conducted to understand the reasons behind the high error rate. Jumping actions typically involve fast and coordinated movements of multiple body parts, including the legs during the jump, changes in posture in the upper body, and body control while airborne. This complexity poses greater challenges for the model in terms of feature extraction and temporal modeling. Several examples of jumping actions are selected to showcase the misclassification instances during the recognition process. **Figure 6** displays key frames of some misclassified actions, with potential causes including similar backgrounds, motion blur caused by the fast speed of the action, and the diversity in action poses. These factors likely contribute to difficulties in feature discrimination, leading the model to struggle with accurate recognition of jumping actions.

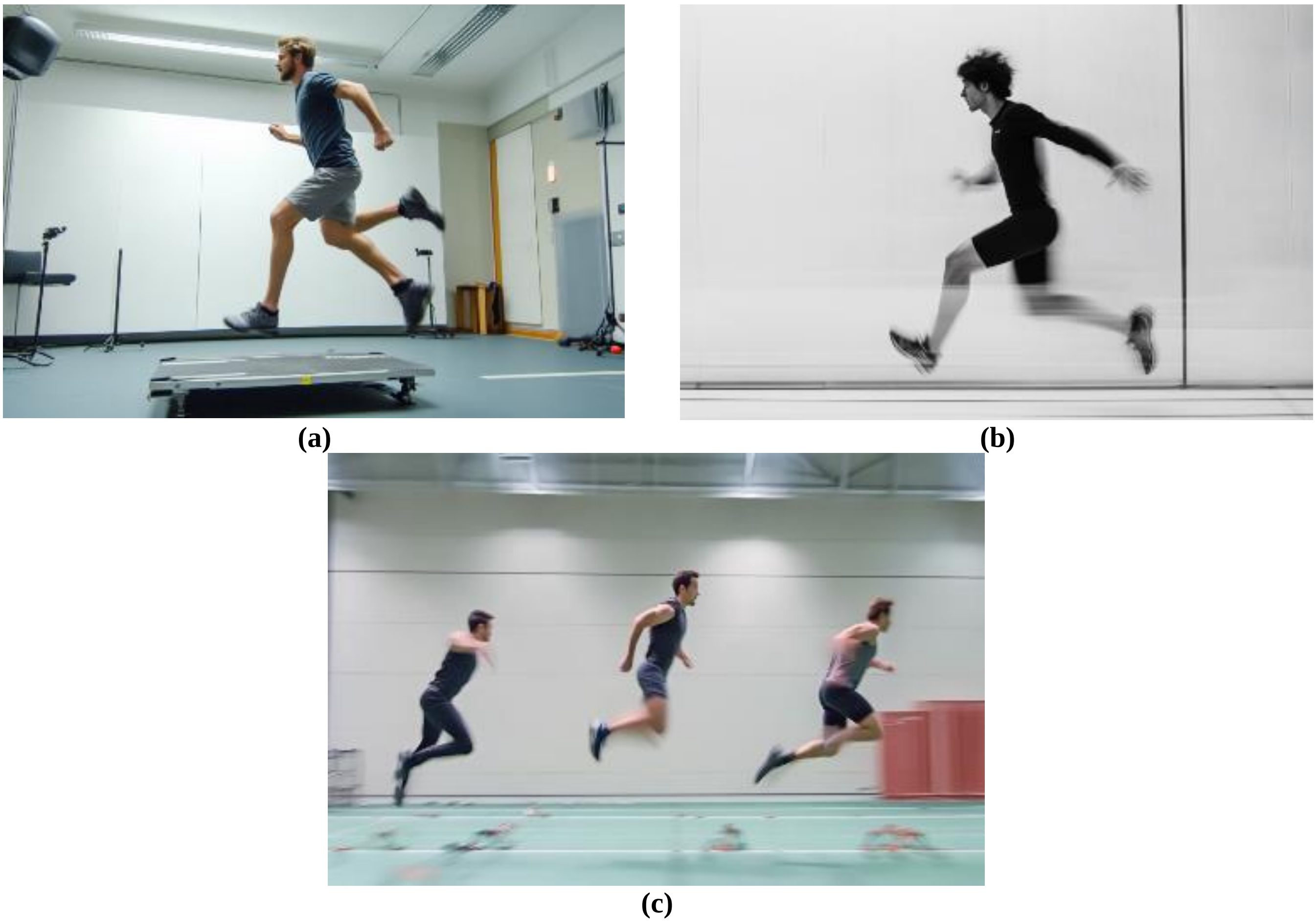


Figure 6. Misclassification examples of jumping actions. **(a)** Movement speed; **(b)** Diversity of action posture; **(c)** Diversity of action posture.

To improve the model’s performance in recognizing jumping actions, several strategies are proposed. Increasing the number of training samples for jumping actions and applying data augmentation with different angles and speeds enhance the model’s robustness to complex motions. Additionally, combining video data with sensor data (such as accelerometers and gyroscopes) for multimodal learning can provide richer motion information, which may help improve the accuracy of jumping action recognition. During model training, fine-tuning hyperparameters, especially in terms of learning rate, batch size, and optimization algorithms, could improve the model’s learning effectiveness. Through an in-depth analysis of the challenges in jumping action recognition, this work aims to provide more targeted guidance for athlete training and technical improvements in practical applications. These findings can contribute to the further advancement of sports action recognition technology.

4.5. Robustness analysis of the model

The performance of the model under different training set sizes is analyzed. **Figure 7** presents the results.

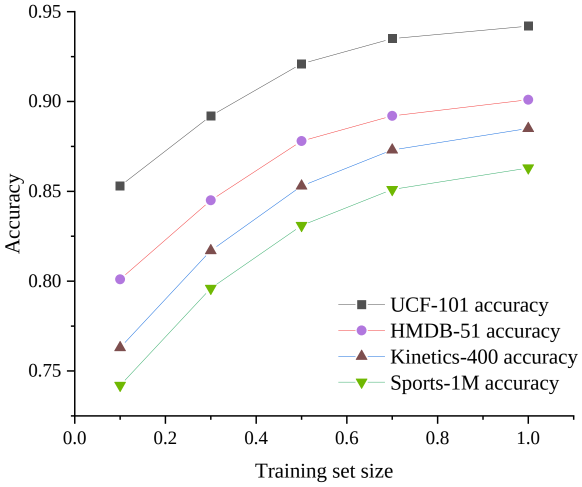
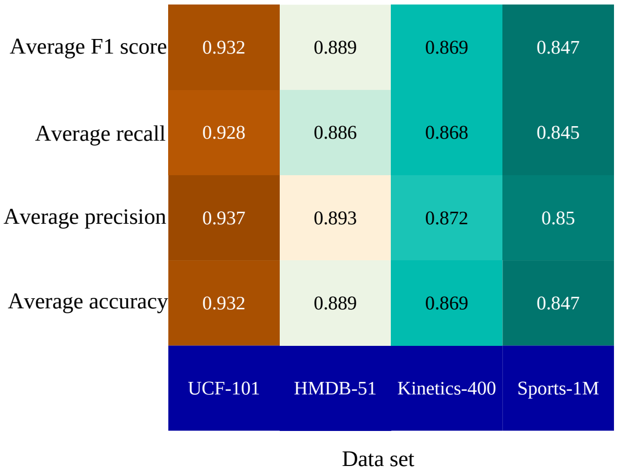


Figure 7. Performance under different training set sizes.

Figure 7 shows that as the training set size increases, the model's accuracy progressively improves. For example, on the UCF-101 dataset, the accuracy increases from 85.3% to 94.2% as the training set size grows from 10% to 100%. This suggests that the model's performance improves with the increase in training data, demonstrating good robustness. Even with a smaller training set, the model still exhibits relatively good performance.

4.6. Model generalization capability

The cross-validation process across different datasets is analyzed. **Figure 8** displays the results.



	UCF-101	HMDB-51	Kinetics-400	Sports-1M
Average F1 score	0.932	0.889	0.869	0.847
Average recall	0.928	0.886	0.868	0.845
Average precision	0.937	0.893	0.872	0.85
Average accuracy	0.932	0.889	0.869	0.847

Figure 8. Cross-Validation results on different datasets.

Figure 8 reveals that the established model maintains high levels of average accuracy, precision, recall, and F1 score across various datasets during cross-validation. For instance, on the UCF-101 dataset, the average accuracy achieved through cross-validation is 93.2%, with other metrics also being very close. The performance on other datasets remains at a high level as well. These results indicate that the proposed model excels not only on individual datasets but also demonstrates strong generalization capability across multiple datasets.

5. Conclusion

This work presents a DL-based athlete skill and motion analysis system. Experiments demonstrate that the system not only effectively identifies various complex athletic actions but also provides valuable feedback for coaches. The results show that the established model achieves high accuracy, precision, recall, and F1 score, with an accuracy exceeding 90% on the UCF-101 dataset, highlighting its significant advantage in action recognition tasks. This work proposes a DL-based athlete action recognition model. Despite its significant performance, there are still several limitations and directions for further exploration.

(1) Model Limitations: While the model performs excellently across multiple datasets, its accuracy may be affected when handling certain complex actions, such as high-intensity dynamic interactions (such as wrestling or fast changes in team sports). Additionally, the model's sensitivity to environmental disturbances may lead to instability in real-world applications. This means that, in real-world scenarios, the model must handle various unpredictable variables, posing challenges for model training and optimization.

(2) Impact on Athlete Training and Competition Analysis: The model offers more accurate action analysis and feedback, assisting coaches in making targeted adjustments during training. For example, by analyzing athletes' technical movements in real-time, coaches can quickly identify deficiencies in the execution of actions and develop personalized training plans accordingly. Additionally, the model can provide data support during competitions, helping coaches with tactical adjustments and opponent analysis.

(3) Practical Application Translation: Implementing the research results in real-world training and competition analysis involves addressing several key challenges. First, the model needs to be integrated into the athlete's daily training monitoring system for real-time analysis and feedback. Second, visualization tools based on model analysis can help coaches better understand an athlete's performance, allowing for timely adjustments. Future research can also explore how to combine this technology with virtual reality or augmented reality to create more immersive and interactive training environments.

When considering the application of this model in real-world settings, several key deployment aspects must be addressed to ensure its feasibility and effectiveness. The model relies on a CNN-LSTM hybrid architecture, which performs well on high-performance computing platforms but optimization of computational resources is still required for deployment on edge devices. Model compression and acceleration techniques, such as quantization and pruning, can reduce computational burden, making it more suitable for real-time processing. The power consumption of edge devices is another crucial factor influencing system deployment. To ensure sustainable operation, it is recommended to consider low-power processors in hardware selection and optimize algorithms to reduce computational demand, ensuring long-duration operation on battery-powered devices. The system latency in real-time processing is crucial for feedback in sports training. By optimizing data transmission and processing pipelines, reducing data transfer time, and improving model inference speed, the overall latency can be effectively reduced. It is suggested to perform latency testing in real-world applications for different sports training scenarios to meet the specific requirements of sports training. This model can be integrated with existing sports training programs to provide real-time feedback to coaches and athletes. By combining with sports monitoring equipment and wearable sensors, the model can analyze athletic performance in real time, offering personalized training recommendations to help athletes optimize their technical movements and training plans. Through discussing these practical deployment considerations, this work aims to provide a comprehensive perspective on the practical application of the research results, enhancing its real-world impact on sports training and analysis.

Author contributions: Conceptualization, MZ and YL; methodology, YC; software, YL; validation, MZ, YL and YC; formal analysis, MZ; investigation, YL; resources, YC; data curation, YC; writing—original draft preparation, MZ; writing—review and editing, YC; visualization, YL; supervision, MZ; project administration, YC; funding acquisition, YC. All authors have read and agreed to the published version of the manuscript.

Ethical approval: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. Cossich VRA, Carlgren D, Holash RJ, et al. Technological breakthroughs in sport: Current practice and future potential of artificial intelligence, virtual reality, augmented reality, and modern data visualization in performance analysis. Applied Sciences. 2023; 13(23): 12965.
2. Abed S A. The relationship between artificial intelligence and sustainability by mediating women’s athletic performance: An exploratory study of the opinions of a sample of women’s sports in the United Arab Emirates Revista iberoamericana de psicología del ejercicio y el deporte. 2024; 19(1): 45–53.
3. Efe A. An Assessment Over The Impact Of Artificial Intelligence On Sports Activities And The Sports Industry Çanakkale Onsekiz Mart Üniversitesi Spor Bilimleri Dergisi. 2023; 6(3): 76–101.
4. Lam W W T, Tang Y M, Fong K N K. A systematic review of the applications of markerless motion capture (MMC) technology for clinical measurement in rehabilitation Journal of neuroengineering and rehabilitation. 2023; 20(1): 57.
5. Werling K, Bianco N A, Raitor M, et al. AddBiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization Plos one. 2023; 18(11): e0295152.
6. Jampani V, Maninis K K, Engelhardt A, et al. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations Advances in Neural Information Processing Systems. 2023; 36: 76061–76084.
7. Kufel J, Bargieł-Łączek K, Kocot S, et al. What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine Diagnostics. 2023; 13(15): 2582.

8. Tapeh A T G, Naser M Z. Artificial intelligence, machine learning, and deep learning in structural engineering: a scientometrics review of trends and best practices Archives of Computational Methods in Engineering. 2023; 30(1): 115–159.
9. Wazirali R, Yaghoubi E, Abujazar M S S, et al. State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques Electric power systems research. 2023; 225: 109792.
10. Li X Q, Song L K, Choy Y S, et al. Fatigue reliability analysis of aeroengine blade-disc systems using physics-informed ensemble learning Philosophical Transactions of the Royal Society A. 2023; 381(2260): 20220384.
11. Alkhudaydi O A, Krichen M, Alghamdi A D. A deep learning methodology for predicting cybersecurity attacks on the internet of things Information. 2023; 14(10): 550.
12. Xu B. Optical image enhancement based on convolutional neural networks for key point detection in swimming posture analysis Optical and Quantum Electronics. 2024; 56(2): 260.
13. Ullah H, Munir A. Human activity recognition using cascaded dual attention cnn and bi-directional gru framework Journal of Imaging. 2023; 9(7): 130.
14. Zan H, Zhao G. Human action recognition research based on fusion TS-CNN and LSTM networks Arabian Journal for Science and Engineering. 2023; 48(2): 2331–2345.
15. Lovanshi M, Tiwari V. Human skeleton pose and spatio-temporal feature-based activity recognition using ST-GCN Multimedia Tools and Applications. 2024; 83(5): 12705–12730.
16. Xin C, Kim S, Cho Y, et al. Enhancing Human Action Recognition with 3D Skeleton Data: A Comprehensive Study of Deep Learning and Data Augmentation Electronics. 2024; 13(4): 747.
17. Ahmed S F, Alam M S B, Hassan M, et al. Deep learning modelling techniques: current progress, applications, advantages, and challenges Artificial Intelligence Review. 2023; 56(11): 13521–13617.
18. Taye M M. Understanding of machine learning with deep learning: architectures, workflow, applications and future directions Computers. 2023; 12(5): 91.
19. Carvalho D D, Goethel M F, Silva A J, et al. Swimming Performance Interpreted through Explainable Artificial Intelligence (XAI)—Practical Tests and Training Variables Modelling Applied Sciences. 2024; 14(12): 5218.
20. Legault I, Faubert J. Gender comparison of perceptual-cognitive learning in young athletes Scientific Reports. 2024; 14(1): 8635.
21. Lee Y H, Chang J, Lee J E, et al. Essential elements of physical fitness analysis in male adolescent athletes using machine learning Plos one/2024; 19(4): e0298870.
22. Li J, Chen D, Qi X, et al. Label-efficient learning in agriculture: A comprehensive review Computers and Electronics in Agriculture. 2023; 215: 108412.
23. Palermi S, Vecchiato M, Saglietto A, et al. Unlocking the potential of artificial intelligence in sports cardiology: does it have a role in evaluating athlete's heart? European Journal of Preventive Cardiology. 2024; 31(4): 470–482.
24. Waheed S R, Rahim M S M, Suaib N M, et al. CNN deep learning-based image to vector depiction Multimedia Tools and Applications. 2023; 82(13): 20283–20302.
25. Al-Selwi S M, Hassan M F, Abdulkadir S J, et al. LSTM inefficiency in long-term dependencies regression problems Journal of Advanced Research in Applied Sciences and Engineering Technology. 2023; 30(3): 16–31.
26. Roumeliotis K I, Tselikas N D. Chatgpt and open-ai models: A preliminary review Future Internet. 2023; 15(6): 192.