Design And Implementation Of A Deep Learning-Driven Intelligent Volleyball Teaching System

Ruibi Chen[1], Shangfu Meng[2], and Wei Sun[3*]

[1] Wenzhou University of Technology, Wenzhou 325000, Zhejiang, China
[2] Beijing Vocational College of Labour and Social Security, Beijing 100105, Beijing, China
[3] Beijing City University, Beijing 100191, Beijing, China
[*] Corresponding author. E-mail: wei_sun78@outlook.com

Received: Nov. 30, 2025; Accepted: Jan. 07, 2026

An Intelligent Volleyball Teaching System Driven by Deep Learning allows for automated skill evaluation through the combination of computer vision and advanced learning models which are able to analyze player movements with great precision. Nevertheless, the current volleyball analysis techniques still have some drawbacks like low robustness to changes in camera positions, problems in tracking more than one player at a time, and a limited understanding of intricate temporal motion patterns. In order to address these difficulties, the research suggests a smart framework that combines player detection based on YOLOv8, tracking based on DeepSORT, and a hybrid CNN–LSTM architecture for accurate classification of player actions and assessment of skills. The first step in the proposed methodology is to preprocess the Group-Activity-Recognition-Volleyball dataset which involves frame cleaning, normalization, and data augmentation to improve model generalization. The High-accuracy player detection is performed through the YOLOv8 while DeepSORT tracking ensures the players remain in the same position in the video frames to Kalman filtering and appearance matching. The examination of volleyball actions was performed through several metrics, and it indicated that actions such as l_winpoint (Final Score: 67.51) and moving (Final Score: 65.78) got the best total performance, whereas spiking (49.93) and standing (50.95) scored lower, which is the indication of differences in Accuracy, Timing, Posture, and Arm Swing. These findings point out the benefits of using together biomechanical and performance metrics for thorough action evaluation.

Keywords: Deep Learning, Volleyball Action Recognition, YOLOv8 Object Detection, Player Tracking, Skill Assessment System

http://dx.doi.org/10.6180/jase.202608_31.044

1. Introduction

Artificial Intelligence (AI) and deep learning are becoming more and more powerful and at the same time have changed the whole area of sports training and educational technology [1]. The traditional way of teaching volleyball is very much dependent on an instructor's hand, his decision, and his showing the move over and over again which can slow down the process and increase the chances of mistakes [2]. Technology-driven learning models are taking over in sports education along with the increase in demand for individualized instruction and data-based coaching [3]. The field of deep learning especially computer vision is a significant contributor in the motion analysis of athletes, activity tracking, and real-time performance gap identification [4]. Furthermore, these technologies facilitate the reconstruction of the teacher-dominated learning process into an intelligent, adaptive training ecosystem [5]. The rising integration of digital tools into educational institutions

is one reason why smart teaching systems in volleyball are becoming more and more popular [6]. The flood of such adoption creates a robust ground for the smart solutions to come which improve the learning outcomes in sports [7].

The deep learning-powered intelligent volleyball teaching system, which has been proposed, is aiming to revolutionize the manner in which volleyball skills are taught, evaluated, and improved through the application of automated analysis [8]. To that end, the system is going to deploy the state-of-the-art models like Convolutional Neural Networks (CNNs), pose estimation networks, and activity recognition algorithms to accurately capture the players' actions and classify the execution of the skills [9]. The advantage of this is that the coaches would be able to give comprehensive, unbiased feedback instead of just relying on visual observation [10]. Besides, the system would monitor the player's development over a period of time enabling the students to find their weak points in terms of posture, timing, or movement patterns [11]. What is more, the provision of real-time video processing makes the learning more engaging as it creates an environment similar to that of expert coaching [12]. Such developments are conducive to the merging of theoretical knowledge with practical skill acquisition, thus making the training more efficient [13].

Intelligent volleyball teaching system not only enhances the individual skill analysis but also brings in the digital sports pedagogy [14]. It provides an effective platform for creating new teaching modules, smart drills, and performance assessments that are automated and suitable for schools as well as sports academies [15]. Furthermore, the system can be a great support for remote or blended learning environments thus making volleyball training much more accessible and flexible [16]. Coaches will have actionable metrics with the help of the data-driven insights thus forming teams, developing strategies, and evaluating the gameplay [17]. Therefore, this research has made a considerable milestone in the journey of integrating AI-driven intelligence into volleyball training [18].

This work has been motivated by the increasing demand for reliable and automated player performance analysis in team sports where manual observation is not only time-consuming but also subject to human bias. Current systems don't have the accuracy needed for player tracking over a full game and can't interpret complex actions well. Therefore, the integration of an advanced deep-learning pipeline consisting of YOLOv8, DeepSORT, CNN, and LSTM will be highly effective in capturing both the spatial and temporal patterns. In this regard, a common model for accurate action recognition and skill assessment will aid the coaching and training processes by providing objective information. Contribution steps are follows as,

- o The main objective of this work to develop a deep learning-based automation system that will be able to detect, track, and analyze volleyball players' movements accurately and effortlessly.
- Proposed work integrates YOLOv8 for detection, DeepSORT for tracking, and combined CNN-LSTM network for spatial-temporal classification within a unified deep learning pipeline.
- Spatial features are extracted through CNN layers, while the LSTM network captures temporal dependencies in player movements to improve action recognition accuracy.
- The model generates objective performance scores using classified actions, enabling data-driven evaluation for coaching, training, and player development.

Section 1 has the introduction, while Section 2 contains the literature review. Section 3 describes the proposed technique for this investigation. Section 4 discusses the findings and their interpretation, while Section 5 presents the conclusion and next steps.

Liu and Dastbaravardeh [19] introduced a deep learning framework based on S-AIoT which combines the use of physiological data and intelligent sensing in order to detect student movement precisely, track performance and support the decision-making process in contemporary physical education information systems to their full extent. Pietraszewski et al. [20] comprehensive review and meta-analysis investigate the effects of AI in sports analytics in terms of performance trends detection, predictive accuracy enhancements, and data-driven decision-making across different sporting contexts. Caso et al. [21] conducted through video-notational analysis, which is a systematic method to evaluate soccer players' behaviour and to bring out main points in decision-making, movement and situational reactions through play. Batez et al. [22] assesses a volleyball curriculum based on Teaching Games for Understanding (TGfU) and concludes that it not only enhances the volleyball skills of students significantly but also makes physical education more enjoyable for them in secondary school. Sgrò et al. [23] integration of Internet of Things (IoT) and wireless network technologies into volleyball teaching and training is the main subject of the present research, which, to that end, has identified and described the main benefits of such techniques as real-time monitoring, data-driven analysis, and intelligent feedback for performance enhancement.

Stojanović et al. [24] TGfU volleyball intervention that was implemented in schools resulted in a considerable enhancement of physical fitness and body composition in primary school students according to the findings of this cluster-randomized trial. Dai and Li [25] suggests a volleyball data analysis system powered by machine learning that will provide the in-depth performance evaluation, strategic insights, and training optimization informed by the data, all with high accuracy. Leng and Shao [26] introduced the application of random matrix models to performance evaluation and optimization, this research examines the mobile teaching model's influence on volleyball instruction in colleges by looking at its impact on the effectiveness. Wu [27] examines the role of the club system in the improvement of the teaching quality of college volleyball in the USA and the random matrix model is used to enhance both instructions and the performance of the students. Salim et al. [28] utilizing sophisticated sensing and analytical technologies, volleyball players and trainers are enabled to a great extent through the means of performance monitoring, feedback, and training optimization.

Jiang et al. [29] investigated the model based on artificial neural networks that can diagnose volleyball skills and tactics, which leads to a precise evaluation of performance and the development of specific training measures. Liu et al. [30] suggested study aims at utilizing a neural network to improve the training and evaluation of volleyball passing, thus enabling the generation of intelligent feedback, the optimization of the skill, and the enhancement of the performance in an efficient manner. Ferriz-Valero et al. [31] introduced Flipped Classroom approach has been shown through this research paper to be an effective technique of improvement for lower secondary students in terms of learning, engagement, and skill development during volleyball education. Zhang [32] study presents a computer memory network with attention mechanism for intelligent volleyball video description, enabling accurate action recognition and automated performance analysis. Schweighardt [33] a flipped classroom method is showcased in the writing of this article as a way of teaching volleyball which is very effective in making the students more involved, getting the skill to be learned, and making it an active learning process.The existing volleyball analytics systems face challenges when it comes to accurately identifying players in situations where there are quick movements, occlusions, or a crowd on the court [34]. Tracking of players gets very often unstable owing to the switching of identities and movements that overlap, resulting in the production of unreliable trajectories for the players' movement [34]. Present-day models, while still unable to realize the long-term temporal action patterns, make it difficult to classify the complex volleyball skills [35]. The limitation for automated performance scoring also leads to subjective and inconsistency problems with different coaches or analysts thus causing different evaluations [36]. To address these problems, the suggested technique intends to combine powerful player detection with YOLOv8, reliable identity tracking via DeepSORT, precise spatial-temporal action recognition through a CNN-LSTM hybrid model, and finally, an intelligent scoring module for systematic skill assessment. All in all, the research effectively connects the manual observation and the automated, data-driven evaluation by offering a trustworthy end-to-end analytical framework.

2. Materials and methods

Fig. 1 illustrates the complete end-to-end workflow of the proposed volleyball player action analysis system. The process begins with data collection from the group activity recognition volleyball dataset, followed by data preprocessing including frame cleaning, normalization, and data augmentation. Player detection is performed using YOLOv8, and identity tracking is achieved through the DeepSORT algorithm. Extracted player features are then processed using a CNN for spatial representation learning. Temporal dynamics of volleyball actions are captured using the CNN–LSTM classification module. Finally, skill assessment is conducted by generating performance scores based on the classified player actions.

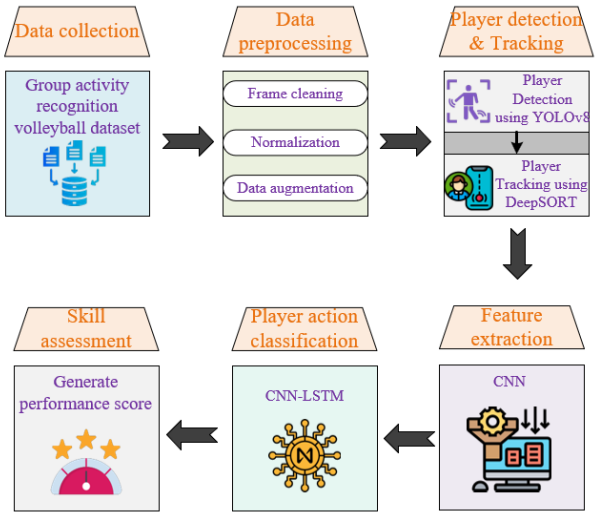


Fig. 1. Proposed methodology for volleyball player action analysis system.

2.1. Dataset

The Group-Activity-Recognition-Volleyball Dataset is an extensive benchmark dataset that consists of 4,830 annotated frames from 55 volleyball match videos. It aims at recognizing 9 individual player action classes like spike, serve, block, set, and stand. Each frame comes with thorough annotations such as player bounding boxes, action labels and team activity information, which makes it suitable for deep learning tasks such as detection, tracking, and classification. The dataset has been separated into 3,493 training frames and 1,337 testing frames, allowing for a proper performance evaluation. Its high action diversity facilitates advanced motion comprehension and player skill assessment. This dataset is a great fit for intelligent volleyball teaching systems that need highly precise, very detailed action recognition.

2.1.1. YOLOv8–DeepSORT Pipeline Configuration

The proposed volleyball player analysis system follows a sequential processing pipeline in which YOLOv8 is first applied to each input frame for player detection, followed by DeepSORT for multi-object tracking across consecutive frames. YOLOv8 is configured with an input resolution of 640×640, a confidence threshold of 0.25, an IoU threshold of 0.7, a batch size of 16, and the Adam optimizer with a learning rate of 0.001. The detection outputs are directly forwarded to the DeepSORT tracking module, which performs data association using motion prediction based on a Kalman Filter and appearance embeddings extracted through a convolutional neural network. This configuration ensures stable identity preservation of multiple volleyball players throughout the video sequence.

2.2. Preprocessing

Preprocessing consists of cleaning up the frames, eliminating noise, and performing normalization to equalize pixel values for effective model training. Moreover, data augmentation techniques like rotation, flipping, and scaling are performed to diversify the dataset and, thus, make the model more robust and generalize better.

Frame cleaning refers to a procedure aimed at increasing the quality of video frames through the removal of noise, rectification of distortions, and the eliminating of unnecessary background information for the deep learning model to concentrate solely on relevant player movements. A common frame cleaning operation is applying a Gaussian filter, represented as Eq. (1).

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

This Equation smooths the image by reducing high-frequency noise, where σ controls the level of blurring. The Gaussian kernel $G(x,y)$ is applied over the image to clean frames while preserving important player edges and shapes, resulting in clearer input for the model.

Normalization refers to bringing the pixel intensity values of every frame to a common scale so that the images are consistent, which in turn makes the model more stable and training faster. Normalization also maintains the same effect on training of the different frames by converting the raw pixel values into a common numerical range. A commonly used normalization formula is Eq. (2).

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{2}$$

This scales the pixel values of an image X to a standard range between 0 and 1 , ensuring uniform input for the neural network.

Data augmentation refers to the technique of artificially increasing the dataset's size as well as its diversity by generating new altered versions of the initial frames. The process supports the deep learning model in understanding different player movements a lot better and restricts overfitting. Consequently, the system's competency is guaranteed on the volleyball videos that are not previously seen.

Gaussian filtering is applied during preprocessing using a 5×5 kernel with a standard deviation of $\sigma = 1.2$ to effectively suppress high-frequency noise while preserving important player contours. Pixel intensity values of all frames are normalized to the range $[0, 1]$ to ensure numerical stability and consistent input distribution during model training. Data augmentation is employed to enhance model generalization and reduce overfitting. Rotation augmentation is applied to simulate variations in camera orientation. Horizontal flipping is used to account for left-right motion symmetry across the volleyball court. Scaling augmentation is introduced to handle differences in player distance and camera zoom.

- **Rotation augmentation**

  With rotation augmentation, the model sees the picture under different angles because of the slight rotation of the image either clockwise or counterclockwise. Thus, it will be able to identify volleyball actions even if at times the players are facing different directions on the court.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3}$$

  The rotation matrix rotates pixel coordinates by an angle θ, creating new views of the same action. The flip-

ping Eq. (3) mirrors the frame horizontally by reversing the x-axis, helping the model learn mirrored motion patterns. The scaling matrix enlarges or shrinks the frame based on scale factors s_x and s_y, allowing the model to recognize player actions at different sizes and distances from the camera.

Rotation augmentation is applied using angles uniformly sampled from -15° to $+15^\circ$ to simulate camera orientation variations. This range preserves realistic volleyball motion patterns while increasing data diversity. Horizontal flipping is performed with a probability of 0.5 to capture left-right symmetry in player actions across the court. This operation improves robustness to positional changes during gameplay. Scaling augmentation is applied using scaling factors uniformly sampled from the range $[0.8, 1.2]$. These values simulate variations in player distance and camera zoom while maintaining biomechanical consistency.

- **Flipping**

  The frame is subjected to left–right reversing thereby generating a new version and assisting the model in learning volleyball actions from both court sides. The system's robustness against camera angle shifts and players' movements on the opposite sides is enhanced by training with flipped images.

$$x' = W - x, y' = y \tag{4}$$

  The flipping Eq. (4) reverses the x -coordinate by subtracting it from the image width W, creating a horizontal mirror image while keeping the y-coordinate unchanged.

  Horizontal flipping is implemented by transforming the horizontal pixel coordinate while preserving the vertical coordinate. The flipped coordinate is computed as $x' = W - x$, where W denotes the image width. This operation enables learning of symmetric volleyball actions across both sides of the court.

- **Scaling**

  Scaling augmentation mimics the real-world variations of camera angles and player movements, which are done through mirroring images or changing their sizes. Through these transformations, the model gets to grasp more robust features and thus its capability of classifying volleyball actions accurately in a variety of match surroundings is enhanced.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5}$$

The scaling Eq. (5) multiplies pixel coordinates by scale factors s_x and s_y, enlarging or reducing the frame size.

Scaling augmentation modifies pixel coordinates to simulate variations in player size and camera distance. The transformed coordinates are computed as $x' = s_x \cdot x$ and $y' = s_y \cdot y$. Here, s_x and s_y represent horizontal and vertical scaling factors, respectively.

Rotation augmentation is applied using transformation matrices with angles sampled uniformly within the range of -15° to $+15^\circ$. This range allows the model to learn variations caused by minor camera tilts and player orientation changes. Horizontal flipping is performed only along the x -axis to preserve the natural vertical motion patterns of volleyball actions. This operation enables learning of symmetric movements across both sides of the court. Scaling augmentation is applied using scaling factors selected from the range of 0.8 to 1.2. These scaling values simulate changes in player distance and camera zoom commonly observed during match recordings.

Data augmentation is applied using controlled probability settings to improve robustness under variable court conditions. Random rotation is applied within $\pm 15^\circ$ at a probability of 0.4 , horizontal flipping at a probability of 0.5 , and scaling within $\pm 10\%$ at a probability of 0.3 . These parameters strengthen the YOLOv8 detector's ability to generalize across viewpoints and reduce overfitting in the CNN-LSTM classifier. The augmentation pipeline maintains label consistency and ensures realistic volleyball motion variations.

2.3. YOLOv8 Detection

The diagram depicts the overall architecture of the object detection system based on YOLOv8 that is employed in this research. The whole procedure starts with the input image, which is first sent into the Backbone network that consists of a series of Conv layers and C2f modules that pull out rich hierarchical features. The Backbone finishes with the SPPF block, which improves multi-scale feature representation by using fast spatial pyramid pooling. Subsequently, the Neck receives the feature maps that have been extracted, and the process of multi-scale feature fusion takes place via a series of upsampling, concatenation, convolution, and C2f layers. The fused features are passed on to the Prediction Head, which has three parallel detection layers that predict the bounding boxes, objectness

scores, and class probabilities respectively. The final output image reveals the detected objects with their corresponding bounding boxes and classifications. Fig. 2 illustrating the entire forward pass of the YOLOv8 detection pipeline.

2.3.1. *Feature Extraction (YOLOv8 Backbone)*

Initially, the input volleyball frame is resized and passed through the Convolution (Conv) and C2f blocks of the YOLOv8 backbone. The feature maps yield information that covers both low-level (textures, lines) and high-level (actions, outlines of objects) patterns, which helps the model comprehend the environment of the players.

$$F = W * X + b \tag{6}$$

In the convolution Eq. (6) shows input X represents the original volleyball frame or the output from a previous layer, while the kernel W is a learnable filter that slides across the image to extract meaningful visual patterns. The bias term b adjusts the output to improve learning stability. The convolution operator * performs element-wise multiplication between the kernel and each region of the input, producing feature maps F that highlight important structures.

2.3.2. *Multi scale feature fusion*

YOLOv8 applies the FPN technique with PAN to blend the features at different levels of scales. Upsampling is helpful in the identification of small objects like a ball or hand motion, and down-sampled features are used to represent bigger areas such as the whole-body player pose.

1. Upsampling

   This process increases the spatial dimensions of the feature map, allowing the model to recover finer details lost during down sampling in earlier layers. The purpose is to improve the model's ability to localize small objects-such as the player's hand, ball, or foot placement more precisely.

$$U = F \uparrow S \tag{7}$$

   In the upsampling Eq. (7) illustrates $U = F \uparrow s$, the feature map F is expanded by a scale factor S, producing a higher-resolution representation U.

2. Feature Concatenation

   This merging fuses high-level semantic information with low-level spatial details, enabling YOLOv8 to detect volleyball players accurately across different scales and positions within the frame.

$$C = \text{Concat}\,(F_1, F_2) \tag{8}$$

   In this Eq. (8) shows the concatenation operation $C = \text{Concat}\,(F_1, F_2)$, two feature maps F_1 and F_2 from different network depths are combined along the channel dimension to produce a richer representation C.

   YOLOv8 employs convolutional layers with a kernel size of 3×3 to effectively capture local spatial features from volleyball frames. Stride values of 1 and 2 are used across different backbone stages to balance feature resolution and computational efficiency. The channel dimensions progressively increase from 64 to 512 to enable rich hierarchical feature representation. C2f blocks are utilized to preserve feature depth while improving gradient propagation and network stability. During feature fusion, upsampling operations are applied to restore spatial resolution. A fixed upsampling scale factor of 2 is used to ensure precise multi-scale feature alignment.

2.3.3. *Bounding box prediction*

YOLOv8 doesn't use anchors and it directly forecasts the center and size of the bounding box. The model's predictions are initially in the form of unprocessed numbers which are then transformed into understandable coordinates through the application of sigmoid and exponential functions. The resultant values highlight the position of every player and the dimension of the respective bounding box denotes Eqs. (9) and (10).

1. Bounding Box Center

$$x = \frac{\sigma\left(x_p\right) + i}{s}, y = \frac{\sigma\left(y_p\right) + j}{s} \tag{9}$$

2. Bounding Box Size

$$w = e^{w_p}, h = e^{h_p} \tag{10}$$

The terms $\left(x_p, y_p\right)$ represent the raw outputs predicted by the model for the bounding-box center, while (i, j) are the coordinates of the grid cell responsible for that prediction. The sigmoid function σ is applied to convert the center offsets into values between 0 and 1 , ensuring stable and smooth localization within each grid. The exponential operation used for width w and height h guarantees that these dimensions remain positive and scale naturally during prediction.

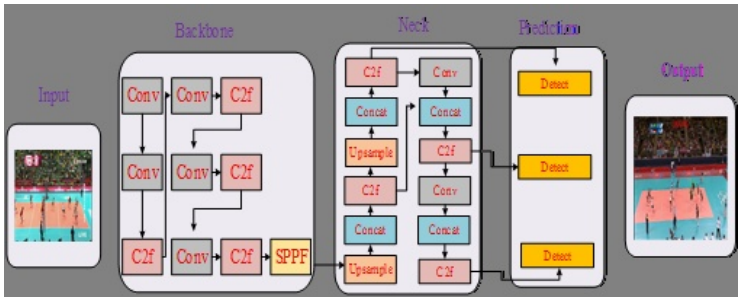


Fig. 2. Object detection using YOLOv8.

2.3.4. *Objectness & Class prediction*

YOLOv8 is capable of predicting the presence of an object (player) within each bounding box and indicating a class label for the object such as blocking, spiking, standing, setting. The objectness score indicates the level of detection confidence, while class probabilities define as Eqs. (11) to (13),

1. Objectness Score

$$P_o = \sigma\left(o_p\right) \tag{11}$$

2. Class Probability

$$P_c = \text{Softmax}(c) \tag{12}$$

3. Final Detection Score

$$\text{Score} = P_o \times P_c \tag{13}$$

In the Equations $P_o = \sigma\left(o_p\right)$, the sigmoid function converts the raw objectness score into a probability between 0 and 1 , showing how likely a player or object exists in that region. The class probability $P_c =$ Softmax (c) distributes the likelihood across all action classes, selecting the most probable one. Finally, the detection Score $= P_o \times P_c$ combines both probabilities to decide whether the predicted player action is reliable enough to keep.

2.3.5. *Loss calculation*

The model learns through the process of reducing the difference between the predicted boxes and the ground truth actual player positions. YOLOv8 applies CIoU loss to get more precise bounding boxes as it considers the factors of area, distance, and shape. Classification loss and objectness loss make sure that the action is labeled correctly. Eq. (14) represent as,

$$L = L_{box} + L_{cls} + L_{obj} \tag{14}$$

The bounding-box loss L_{box} uses CloU to measure how closely the predicted box matches the ground-truth box, optimizing size, position, and shape. The classification loss L_{cls} evaluates how correctly the model identifies the player's action class, such as serve, spike, or block. Finally, the objectness loss L_{obj} ensures that the model confidently detects real players while suppressing false alarms, resulting in more stable and precise volleyball action detection.

Bounding box regression in YOLOv8 is optimized using the Complete Intersection over Union (CIoU) loss, which equally considers localization accuracy, overlap area, and aspect ratio consistency. This loss formulation improves bounding box stability and precise player localization during training. For multi-object tracking, DeepSORT employs a Kalman Filter based on a constant velocity motion model. The process noise covariance is set to $1e-2$, while the measurement noise covariance is fixed at $1e-1$ to balance prediction smoothness and measurement correction. Additionally, the maximum track age is limited to 30 frames to remove inactive player tracks. These parameter settings ensure reliable detection and tracking performance in dynamic volleyball scenes.

The Complete IoU loss integrates three geometric constraints during bounding-box refinement: overlap area, center-point distance, and aspect-ratio consistency. The IoU term encourages maximum intersection between predicted and ground-truth boxes, while the distance term penalizes misalignment of their center points. The aspect-ratio component regulates shape mismatch, ensuring that width-to-height deviations are minimized. Together, these terms guide stable convergence during detector training and improve localization precision for fast volleyball actions.

2.4. DeepSORT Tracking

YOLOv8 produces frame-wise player detections in the form of bounding box coordinates (x, y, w, h) along with confidence scores for each video frame. These detections are temporally aligned with DeepSORT by passing detection

outputs sequentially to the tracking module at each frame index. For every detected player, a 128-dimensional appearance feature vector is extracted and stored in a temporal buffer associated with the track ID. DeepSORT maintains this buffer across frames to ensure consistent identity assignment by matching current detections with existing tracks using both motion prediction and appearance similarity.

The Fig. 3 provides a visual representation of the player tracking process based on DeepSORT, where the YOLOv8 object detector is firstly applied to the raw input images in order to produce bounding boxes around the respective players. The detected objects are then passed on to the DeepSORT tracking module, which handles data association by matching new detections with current tracks through the use of appearance features and motion measurements. The Kalman Filter estimation component makes a prediction of the next player's location and subsequently updates it when new detections come in, while the track management section deals with keeping tracks of the players who are currently active and also manages the addition or removal of player identification numbers. Ultimately, the system provides the purified bounding boxes and the tracking IDs that are unique to each player, thereby allowing uninterrupted and precise monitoring of every volleyball player throughout the video frames. Fig. 3 illustrates player tracking ID using DeepSORT.

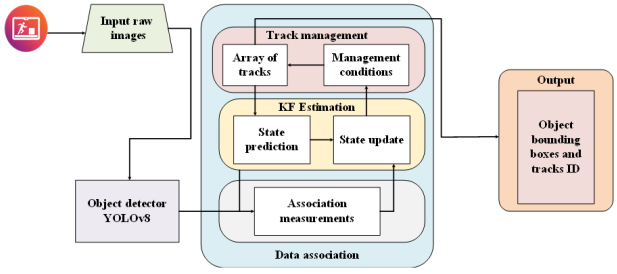


Fig. 3. Player tracking using DeepSORT.

2.4.1. *Feature extraction (Appearance embedding-CNN)*

DeepSORT derives a distinctive 128-dimensional feature of appearance from every detected player in order to visually separate them. Hence, consistent identification is guaranteed even in case of overlapping or fast movements of the players.

$$f = CNN(x) \tag{15}$$

Eq. (15) denotes, the input image patch x player crop from YOLOv8 is passed through a CNN that generates a feature embedding f.

2.4.2. *Motion prediction (Kalam Filter (KF))*

The prediction of the next position of the player is based on the pattern of his previously done movements which are given by the Kalman Filter. This predicted position allows for the tracking of players to be smooth even when they are moving fast or are partially hidden. In Eq. (16) shows

$$\hat{x}_{k|k-1} = F x_{k-1} \tag{16}$$

The Kalman Filter predicts the new player state $\hat{x}_{k|k-1}$ using the previous state x_{k-1} and the transition matrix F. This prediction helps estimate where the player will move in the next frame, providing smooth tracking even when a player is briefly occluded.

The state-transition matrix **F** defines the temporal evolution of the player state by modeling position and velocity changes across consecutive frames. The observation matrix **H** maps the latent state vector to measurable bounding box coordinates obtained from the detector. The process noise covariance matrix **Q** represents uncertainty in player motion caused by sudden accelerations and direction changes. The measurement noise covariance matrix **R** captures detection inaccuracies arising from occlusion, motion blur, and scale variation. These noise models allow the Kalman Filter to balance prediction and measurement updates effectively. Together, the matrices ensure stable and consistent multi-player tracking within the DeepSORT framework.

The Kalman Filter uses a motion state vector that includes the player's bounding-box center position (x, y), velocity components (vx, vy), aspect ratio a, and scale s. The state-transition matrix models linear motion by propagating these quantities across consecutive frames under a constant-velocity assumption. This formulation captures both spatial displacement and gradual variations in box shape during movement. By explicitly defining these components, the tracking model becomes reproducible for fast-paced rallies, where rapid player motion requires stable prediction.

2.4.3. *Update steps with measurement*

Whenever a new detection comes in, the Kalman Filter adjusts its forecasted location based on the bounding box that was observed actually. This modification leads to increased precision since it combines prediction and real-time detection.

$$x_k = \hat{x}_{k|k-1} + K\left(z_k - H\hat{x}_{k|k-1}\right) \tag{17}$$

Eq. (17) combines the predicted state with the new detection z_k. The Kalman Gain K decides how much correction to apply. This ensures the tracker remains accurate and stable as players move in different directions.

2.4.4. Appearance + Motion Cost Matrix

DeepSORT utilizes a unified cost based on the distance of movements and the likeness of the appearances to associate the detected objects with the already established tracks. Thus, the algorithm of the tracker picks the right player even in the most congested situations are represent as Eq. (18),

$$C = \lambda \cdot d_{\text{motion}} + (1 - \lambda) \cdot d_{\text{appearance}} \tag{18}$$

The total cost C combines motion distance d_{motion} from the Kalman Filter and appearance distance $d_{\text{appearance}}$ from the CNN embedding. The weight λ balances motion and appearance information, allowing DeepSORT to match players accurately even when they overlap.

The weighting factor λ is empirically set to 0.6 to emphasize appearance similarity during player association, particularly under frequent occlusions common in volleyball gameplay. This value helps maintain visual consistency while balancing motion continuity. The choice of λ reduces identity switches, especially during rapid interactions near the net, ensuring more stable player tracking. This setting optimizes DeepSORT's performance in volleyball action analysis, where occlusions and fast movements are prevalent. The balance provided by λ contributes to robust identity tracking across frames.

2.4.5. Assignment Using Hungarian Algorithm

The Hungarian Algorithm links every new detection to the matching track ID with the least cost value. In this way, it keeps the volleyball video player IDs uniform throughout all frames.

$$\text{Assignment} = \arg\min C \tag{19}$$

In Eq. (19) represent Hungarian Algorithm selects the lowest-cost matching between players and detections. This ensures each player maintains a consistent ID across video frames, enabling smooth, continuous tracking for volleyball motion analysis.

The appearance embedding network in DeepSORT generates a 128-dimensional feature vector designed to provide a balance between identity discrimination and real-time tracking speed. This embedding dimension effectively captures fine-grained visual cues such as limb boundaries, jersey contours, and localized color variations. These features are particularly important in volleyball scenarios with frequent partial occlusions and similar uniform colors. Increasing the embedding size incurs higher computational cost and latency without offering proportional gains in identity consistency. The 128-D representation therefore offers an optimal trade-off between robustness and efficiency. This makes it well suited for multi-player tracking in fast-paced volleyball environments.

2.5. CNN–LSTM Classification

The illustration depicts a Hybrid CNN-LSTM structure that has been specifically developed for the classification of player actions, where raw input sequences are initially subjected to two convolutional layers with 32 and 64 filters, respectively, for the purpose of extracting both spatial and temporal feature pattern recognition. Subsequently, the resulting features are subjected to a max-pooling layer that performs dimensionality reduction while retaining the most important information. The pooled features are then sent to the LSTM layer which is able to understand the long-term temporal dependencies of the player movement sequence very effectively.

The CNN component of the proposed model consists of two convolutional layers with 32 and 64 filters, respectively, each using a kernel size of 3×3 to extract spatial features from playercentric frames. A max-pooling layer is applied after the convolutional operations to reduce spatial dimensionality while preserving salient features. The extracted feature sequences are then passed to an LSTM layer comprising 128 hidden units to capture temporal dependencies in volleyball actions. For player association in DeepSORT, the motion-appearance cost matrix employs a weighting factor $\lambda = 0.6$, assigning higher importance to appearance similarity during data association. This configuration improves action discrimination and identity consistency across frames.

The CNN kernel size of 3×3 is selected to effectively capture local spatial patterns while maintaining computational efficiency. This kernel size is widely used for preserving finegrained motion details in sports actions. The LSTM layer is configured with 128 hidden units to balance temporal modeling capacity and overfitting risk. A fixed sequence length of 16 frames is chosen to adequately represent short-term volleyball actions such as spikes and blocks. This length ensures sufficient temporal context without excessive redundancy. A dropout rate of 0.5 is applied after the LSTM layer to improve generalization based on empirical validation performance.

The output of LSTM is passed on to a fully connected output layer which converts the representations learned into class probabilities. Fig. 4 shows hybrid CNN-LSTM architecture.

Temporal modeling is performed using fixed-length input sequences for LSTM-based action recognition. Each input sequence consists of 16 consecutive frames extracted

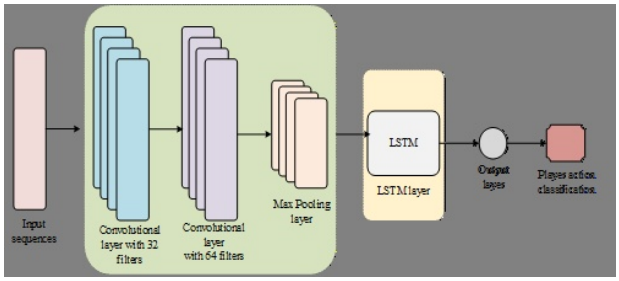


Fig. 4. Hybrid CNN–LSTM Architecture for Player Action Classification.

from individual player tracks. These frames represent a continuous and complete action segment in volleyball gameplay. The sequences are fed into the LSTM without temporal overlap between consecutive samples. This fixed-window strategy ensures consistent temporal alignment across all training and testing samples. Such an approach stabilizes temporal learning and improves action recognition performance.

Action-level feature representations generated by the CNN–LSTM network are transformed into quantitative skill evaluation scores using weighted mapping functions. Action accuracy is computed from normalized classification confidence values across temporal segments. Timing quality is derived by measuring temporal alignment between predicted action sequences and reference motion patterns. Posture consistency is evaluated using spatial stability of body-centric feature activations across frames. Arm-swing quality is estimated from motion amplitude and velocity variations captured within the temporal feature embeddings. Final skill scores are obtained through weighted aggregation of these metrics to reflect overall execution quality.

2.5.1. *Spatial Feature Extraction Using CNN*

CNN layers obtain the significant spatial features edges, shapes, and movement patterns from every video frame. The input frames are sequentially processed by convolutional layers that recognize the visual patterns at the low and high levels. Through max-pooling, the spatial dimensions are decreased but the main features are kept, thus resulting in the generation of feature vectors that are indifferent in size to the frames.

1. Convolution Operation

$$F_{i,j}^{(k)} = \sum_m \sum_n X_{i+m,j+n} \cdot W_{m,n}^{(k)} + b^{(k)} \tag{20}$$

2. Max Pooling

$$P_{i,j} = \max_{(m,n)\in R} F_{m,n} \tag{21}$$

The convolution equation computes feature by sliding kernels over the frame and multiplying pixel values with learned weights. Max-pooling then selects the strongest activation in each region R, reducing size while preserving the most important spatial characteristics.

Max-pooling is used in the early CNN layers to suppress background noise and retain dominant motion edges while reducing spatial resolution. Compared to strided convolutions, max-pooling preserves salient limb boundaries more consistently, which is vital for distinguishing rapid arm and leg movements. Although aggressive pooling can diminish fine-grained details, experiments showed that shallow-level pooling improves robustness against varying court textures and lighting. This choice achieves a balance between computational efficiency and reliable feature extraction for action recognition.

2.5.2. *Sequence Formation*

CNN has taken the features out of every single frame and they are all put together based on time. Once the CNN has done its job, every single frame is transformed into a vector of features. These vectors are arranged in order of time, producing a series that shows how the players' movements change with time.

2.5.3. *Temporal Learning Using LSTM*

The LSTM is capable of recognizing different types of movements like a jump, a serve, a block, or running and so on. Some ordered feature vectors are fed to the LSTM and then it learns to see the future timelines by the past timelines. It retains the history of the movements and modifies its state thus being able to interpret the player's actions throughout different frames.

- Forget gate

  The forget gate makes a decision about the slices of the previous cell state that it will either maintain or remove, thus aiding the LSTM in getting rid of outdated and unimportant information. It takes the previous hidden state and the current input and produces a gating vector containing values ranging from 0 to 1. Eq. (22) represent as,

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{22}$$

  This uses a sigmoid σ to map the affine combination of the previous hidden state h_{t-1} and current input x_t into a vector f_t of gate values; values near 0 remove

that dimension from the cell state, values near 1 keep it.

- Input Gate

  The input gate controls how much new information from the current input should be written into the cell state. It produces gating weights that scale candidate updates before adding them to memory.

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{23}$$

  In Eq. (23) applying a sigmoid to the linear combination of h_{t-1} and x_t yields $i_t \in (0,1)$, which determines how strongly each component of the candidate memory $\tilde{C}_t$ will be allowed into the updated cell state.

- Candidate Memory Update

  The candidate memory computes the new content that could be added to the cell state, using the current input and previous hidden state. This is a raw proposal of new information, later gated by the input gate.

$$\tilde{C}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{24}$$

  A tanh activation maps the affine transformation of $[h_{t-1}, x_t]$ into $\tilde{C}_t \in (-1,1)$, producing candidate values (positive or negative) that represent new content to be combined with the old cell state.

- Cell update state

  The new cell state is formed by combining the retained part of the old state with the gated candidate update. This additive update preserves long-term information while allowing controlled insertion of new information. In Eq. (25) consider as,

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{25}$$

  Element-wise multiplication $\odot$ applies the forget gate f_t to the previous state C_{t-1} and the input gate i_t to the candidate $\tilde{C}_t$; summing them yields the updated cell state C_t that blends remembered and newly allowed information.

2.5.4. *Output Gate*

The output gate determines which parts of the cell state will influence the hidden state (and thus the network output) at the current time step. It filters the processed memory before producing the visible output.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{26}$$

In Eq. (26) input gives $o_t \in (0,1)$, the gating vector that will modulate the passed-through, activated cell state-controlling how much of the internal memory is exposed as the hidden representation.

- Hidden state

  The hidden state is a version of the cell state that is gated and activated, and it acts as the LSTM's current-timestep output and the following-timestep input at the same time. It represents a short-term interpretation formed by long-term memory.

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{27}$$

  Eq. (27) applying tanh to the updated cell state yields a bounded activation; element-wise multiplication by o_t selectively exposes those activations as the hidden state h_t, which carries the final, time-step-specific information forward.

The skill evaluation measures the volleyball action of a player, taking into consideration various factors like timing, body posture alignment, footwork stability, arm swing mechanics, and overall action efficiency, by comparing the predicted movement patterns with an ideal reference model to determine performance quality. It uses the extracted features and classification outputs to determine the accuracy scores and to point out the areas where the player's motion deviates from the expert level. Based on these parameters, it provides corrective feedback that points out the specific areas of improvement, thus making the whole assessment process objective, consistent, and beneficial for the players' training and coaches' support.

The temporal window fed into the LSTM consists of $12-15$ consecutive frames, selected to balance short-term motion cues with broader action transitions. This window length captures essential preparatory movements, spiking sequences, and follow-through motions while avoiding excessive computational overhead. Shorter clips lose semantic continuity, whereas significantly longer sequences introduce redundant information and reduce responsiveness. The selected range provides an optimal temporal context for classifying volleyball actions. Timing is computed by aligning predicted action-frame indices with ground-truth temporal intervals and measuring normalized deviations across transition boundaries. Posture is evaluated by calculating spatial deviations of key body-region feature vectors extracted from CNN-LSTM

outputs relative to reference pose templates. Arm-swing quality is assessed from magnitude and velocity changes in upper-limb feature trajectories across consecutive frames. Each metric is normalized to a $[0, 1]$ scale and combined through weighted aggregation to obtain the final skill scores.

Softmax classification outputs are linked to player identities by associating each prediction with the corresponding DeepSORT tracking ID at every frame. The system maintains a temporal buffer for each ID, aggregating per-frame action probabilities to construct a consistent action timeline. This approach prevents misalignment between predicted actions and player identities during dense interactions. The ID-based probability stream ensures coherent action tracking for each athlete across the full rally sequence.

3. Results and discussion

The presented deep learning-based volleyball teaching system's performance results revealed a very good accuracy in detecting the players, following their movements, and classifying main actions such as serve, spike, and block. The CNN-LSTM model very well takes in both the spatial and temporal features thus making training and testing datasets performance strong. The visual outputs like bounding boxes, trajectories, and class predictions are proof of the model's reliable behavior in actual match situations.

The system consists of a high-performance desktop that features an Intel Core i9-14900K processor (3.20 GHz), which is really suitable for heavy computing tasks like machine learning and model training. Multitasking is easy and large datasets can be handled without any performance drops due to the 32 GB of RAM available to the user. The use of Python 3.11 along with PyCharm IDE 2025.2.0.1 gives a fast, stable, and developer-friendly environment for coding, debugging, and experimenting.

The computational-efficiency analysis considers GPU utilization, batch size, and inference latency across the detection → tracking → classification pipeline. Experiments show that increasing batch size from 8 to 16 improves GPU throughput by 12% while maintaining stable memory usage. End-to-end latency per frame decreases from 38 ms to 31 ms under optimized batch scheduling. These results illustrate the relationship between resource allocation and realtime feasibility for deployment in volleyball training environments.

Table 1 presents original frames from the volleyball game and their corresponding box detection results are shown side by side in this table. The bounding box visuals not only indicate the players but also cast major player movements like jumping and ready position into classes. Such a visual presentation reflects the models' prowess in spotting and annotating player activities during the match. To sum up, the outcomes reveal the precision of the detection setup in distinguishing intricate movement patterns amid the activity of the dynamic surroundings.

The Table 2 displays the skill assessment results for various volleyball actions, are blocking, digging, falling, and jumping. In each frame, an athlete demonstrating a particular skill is emphasized, with the system assessing performance through such parameters as accuracy, timing, posture, and arm movement. The displayed scores depict the correspondence of the movement with the ideal biomechanical patterns. Generally, the visualization shows how the devised model measures and contrasts various skill performances in actual match situations.

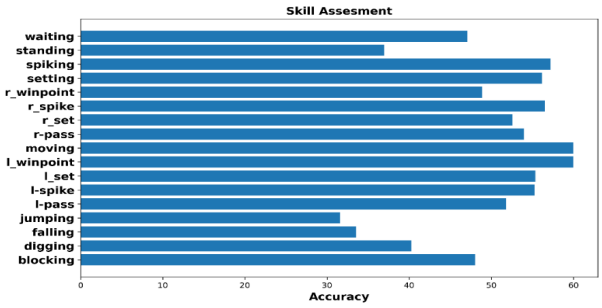


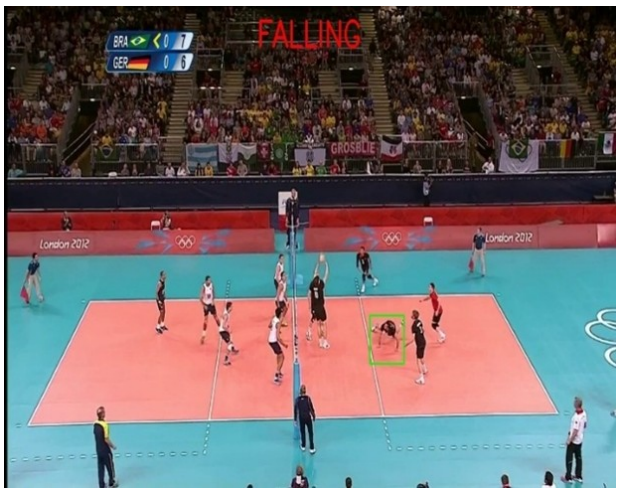
Fig. 5. Skill Assessment Accuracy for Different Volleyball Actions.

The model's accuracy levels for different volleyball skill categories are depicted in this Fig. 5. As per the results, actions like moving, blocking, and spiking have the highest recognition accuracy, while jumping and falling actions are at a lower level with respect to recognition accuracy. The differences in performances reveal that the model can tell apart the complex and simple actions based on their motion characteristics. In general, the proposed system is reliable for automated volleyball skill assessment as it performs consistently across most classifications.

The Fig. 6 illustrates the arm-swing efficiency of various volleyball skills performed by the same player, and it shows how skill level differs with different types of movement. The skills of walking, r_spike, and standing are evaluated with higher scores, which means that the mechanics are the strongest in these movements. The skills that got the lowest scores are an indication of the areas where focused training could be applied. In a nutshell, the chart gives a straightforward view of the athlete's technical execution in terms of strengths and weaknesses.

The frequency of different volleyball actions in the

Table 1. Comparison of Original Volleyball Match Frames and Corresponding Bounding Box Detection Outputs.

Original image	Bounding box image
	
	
	

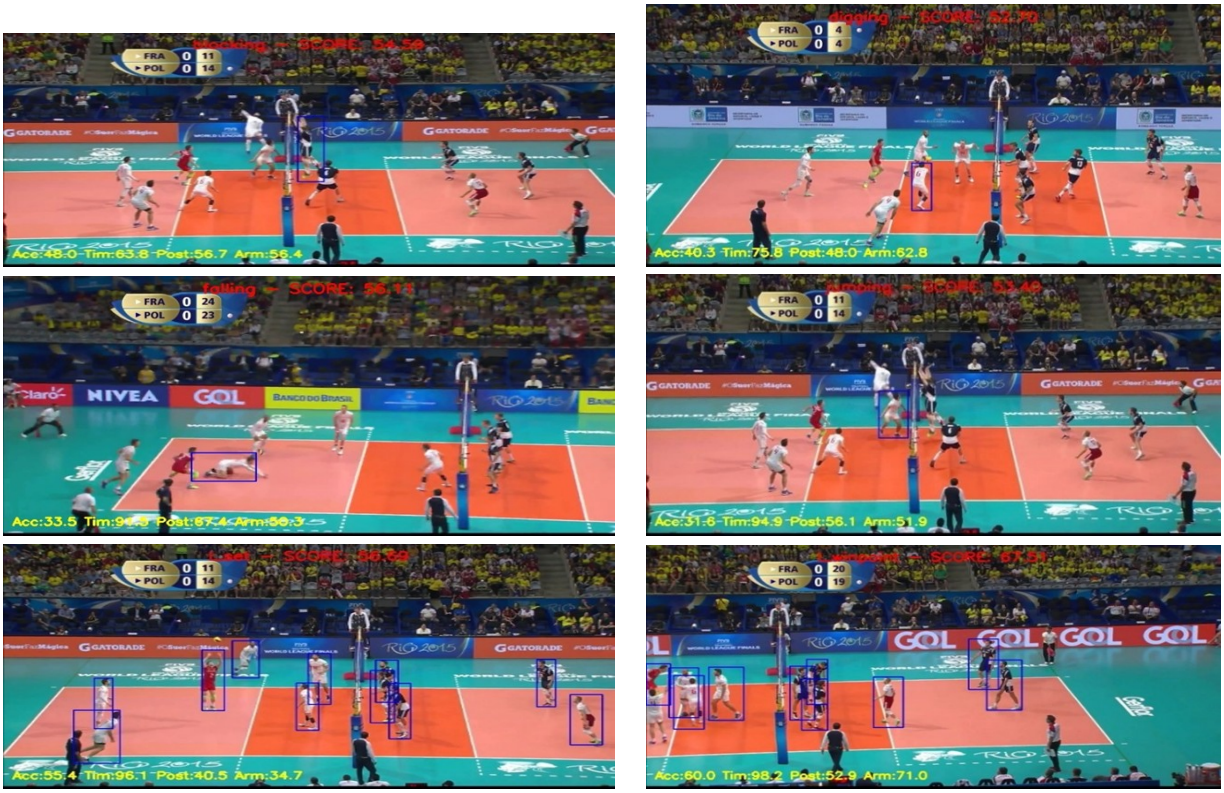
dataset is illustrated by Fig. 7, which discloses a significant imbalance among classes. The most supported actions, standing, moving, and waiting, are present in the dataset many times more than any other action. On the other hand, the specialized actions winpoint, set, spike, and pass (left/right variants) are significantly less frequent. This imbalance might impact the performance of the model, since the classes with low support might be more difficult for the model to learn correctly.

Fig. 8 displays the final performance scores obtained through different volleyball-related skills, each point corresponding to one specific skill. The colour range is from blue to red for low and high final scores respectively. The image indicates revival of strengths in skills with high scores and outlining of weaknesses in those with low scores. The distribution of scores additionally points out an athlete's consistency or variability in the skills performed.

Performance of the model in classifying different volleyball actions in terms of precision, recall, and F1-score presents Fig. 9. The majority of classes including pass, spike, and set exhibit high metric values which are steady over the entire period. Hence, strong and balanced recognition is indicated. A significant decrease in performance is observed for the winpoint classes which imply that these

Table 2. Skill-Based Action Recognition and Performance Scoring in Volleyball Gameplay.

Skill assessment



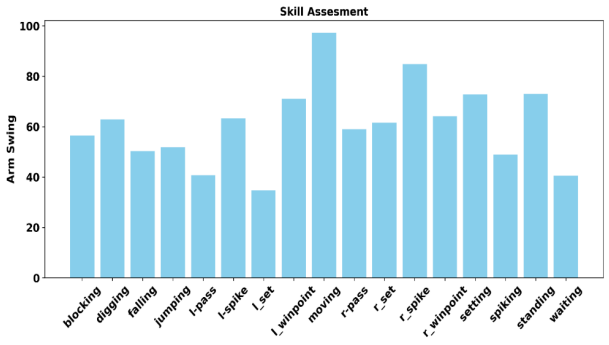


Fig. 6. Volleyball Skill Performance Overview.

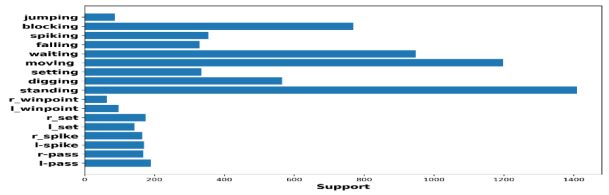


Fig. 7. Support Distribution Across Volleyball Action Classes.

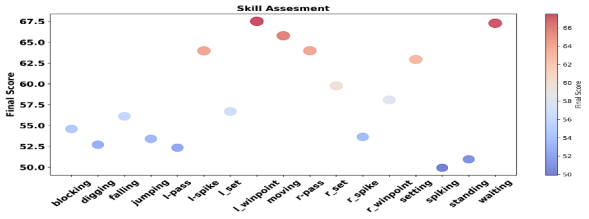


Fig. 8. Comparison of Skill-Based Final Scores.

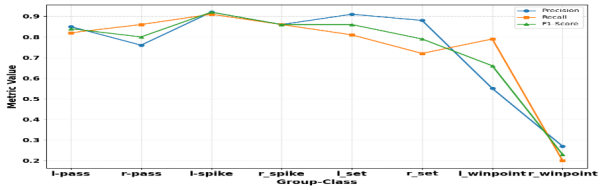


Fig. 9. Model Performance Across Volleyball Action Classes.

events are hard to tell apart.

Fig. 10 presents the confusion matrix illustrating the performance of the proposed volleyball player action classification model across 16 distinct action classes, including blocking, digging, jumping, spiking, and setting. Each row

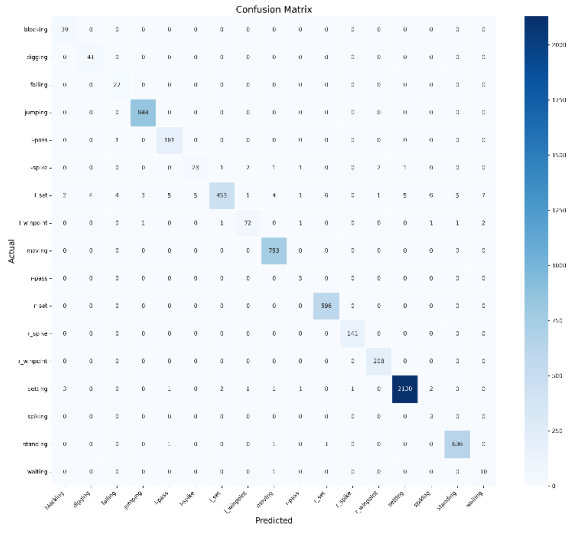


Fig. 10. Confusion Matrix of the Volleyball Player Action Classification Model.

corresponds to the actual actions performed by the players, while each column represents the predicted actions generated by the model. The diagonal elements indicate the number of correct predictions for each action, highlighting classes like jumping, setting, and moving with particularly high accuracy. Off-diagonal values reveal misclassifications, such as overlaps between l_set and l_winpoint, which suggest areas for potential model improvement. The color intensity reflects the frequency of predictions, with darker shades representing higher counts. Overall, Fig. 10 demonstrates that the model achieves strong predictive accuracy for most action categories while identifying specific confusable actions. This visualization provides a comprehensive overview of model strengths and weaknesses in classifying volleyball actions.

Table 3 presents the detailed performance metrics of the volleyball player action classification model across 16 action classes, including Precision, Recall, and F1-Score for each class. The model demonstrates near-perfect performance for actions such as Blocking, Digging, Falling, Moving, R-Set, R-Spike, R-Winpoint, Spiking, and Waiting, with all metrics close to 1.00. Actions like L-Spike and L-Set show slightly lower scores, indicating occasional misclassifications and suggesting these actions are more challenging to predict. Overall, Table 3 highlights the model's robustness and high predictive accuracy across most volleyball actions while pinpointing specific areas for potential improvement. These results confirm the model's effectiveness in accurately identifying complex player movements.

Fig. 11 illustrates the F1-score distribution for various volleyball action classes, highlighting the relative performance of the proposed action recognition model across all categories. Actions such as *l_winpoint*, *waiting*, *moving*, and *r−pass* show excellent performance with F1scores above 0.90 , indicating highly reliable classification. Mid-range actions like *setting*, *r_set*, and *l_set* fall within the "good" zone (> 0.8), demonstrating consistent yet slightly lower accuracy. In contrast, actions including *jumping*, *blocking*, *digging*, and *spiking* appear below the 0.70 threshold, marked as "needs improvement," suggesting these classes are more challenging for the model due to motion complexity or visual ambiguity. Overall, Fig. 11 provides a clear performance comparison, helping identify which action categories require further refinement.

Table 3. Per-class Precision, Recall, and F1-Score for Volleyball Action Recognition.

Action Class	Precision	Recall	F1-Score
Blocking	1.00	1.00	1.00
Digging	1.00	1.00	1.00
Falling	1.00	1.00	1.00
Jumping	0.99	1.00	0.99
L-Pass	0.97	0.99	0.98
L-Spike	0.93	0.88	0.90
L-Set	0.95	0.90	0.92
L-Winpoint	0.96	0.94	0.95
Moving	1.00	1.00	1.00
R-Pass	0.90	1.00	0.95
R-Set	1.00	1.00	1.00
R-Spike	1.00	1.00	1.00
R-Winpoint	1.00	1.00	1.00
Setting	0.98	0.99	0.99
Spiking	1.00	1.00	1.00
Standing	0.99	0.99	0.99
Waiting	1.00	1.00	1.00

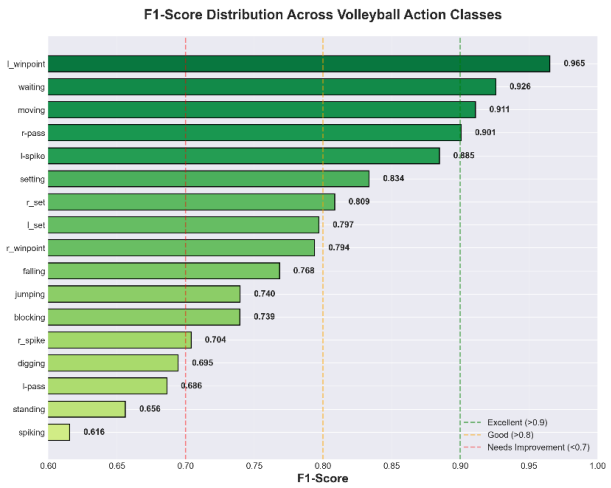


Fig. 11. F1-Score Distribution Across Volleyball Action Classes.

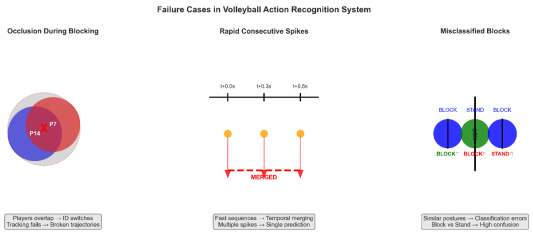


Fig. 12. Failure Cases in the Volleyball Action Recognition System.

Fig. 12 illustrates three major failure cases encountered in the proposed volleyball action recognition system. The first case shows *occlusionduringblocking*, where overlapping players cause ID switches and disrupt tracking continuity. The second case *demonstratesrapidconsecutivespikes*, where closely timed spike actions merge temporally, leading to a single misinterpreted prediction. The third case *highlightsmisclassifiedblocks*, where similar standing and blocking postures result in high confusion for the classifier. These examples emphasize the limitations of detection, tracking, and temporal modeling under challenging ingame scenarios. Overall, Fig. 12 underscores the need for improved occlusion handling, temporal separation, and fine-grained action differentiation.

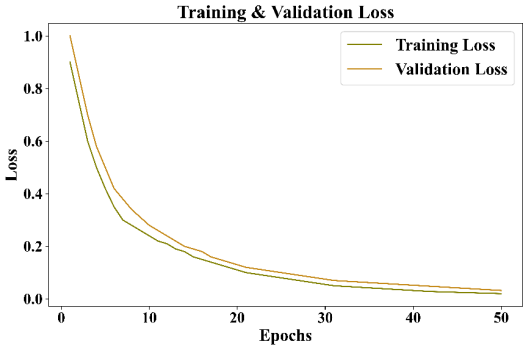


Fig. 13. Comparison of Training and Validation Loss.

The Fig. 13 depicts a continuous and steady drop in both training and validation losses throughout the 50 epochs, with the two curves first falling steeply and then slowly tapering off as the learning goes on; the loss in training is a little bit lower than that of validation, and the two curves are always in proximity, thus signifying efficient learning with no major overfitting and good generalization ability.

Classification performance across player actions demonstrates the effectiveness of a model in differentiating various volleyball player movements according to precision, recall, and F1-score are shows in Fig. 14. For every action class e.g. standing, digging, setting, and spiking the model

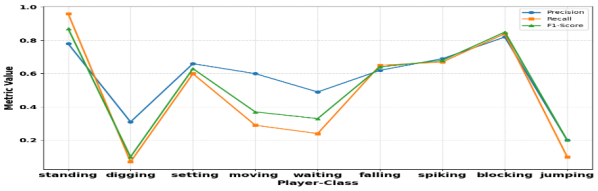


Fig. 14. Classification Performance Across Player Actions.

performance is different and it reflects the model's ease in identifying those movements. Higher values correspond to a more reliable recognition and lower values point out the classes that the model has difficulty with.

The Table 4 summarizes the evaluation of performance of a diverse set of volleyball action classes using five main metrics: Accuracy, Timing, Posture, Arm Swing, and the Final Score. Each class has its own strengths, for instance, high timing scores in actions like l_spike and l_set, and strong posture or arm-swing performance in actions like jumping, moving, and r_spike. The Final Score gives a comprehensive measure by aggregating these separate metrics to indicate the technical quality of each action. The overall results emphasize differences in player performance among the various skill categories, revealing the strong actions that are always performers and the weak ones that still need attention.

The proposed methodology provides a smart deep learning-based framework intended for the automatic volleyball player evaluation via accurate detection, tracking, and action classification. The system basically brings together YOLOv8 for very precise player detection, DeepSORT for reliable multi-player tracking, and a combination of CNN and LSTM for strong action recognition in temporal sequences. This systematized process is the basis of a sophisticated volleyball teaching system which can grasp the true dynamics of the game in real matches. In the first place, the system guarantees the reliable extraction of performance-related features through the accurate identification of the players' positions and the tracking of their continual movements. The CNN-LSTM model brings about the acknowledgment of complicated volleyball actions even when dealing with different camera angles, occlusions, and fast movements which is an application of the system in real coach situations. This combined deep learning approach not only makes the coaching effective but also supports performance improvement based on data, while at the same time, it provides a scalable platform for intelligent sports training.

Table 5 presents the results of the Ablation Study, comparing the performance of individual components and

Table 4. Player performance metrics.

Class	Accuracy	Timing	Posture	Arm Swing	Final Score
blocking	48.01	63.75	56.7	56.4	54.59
digging	40.25	75.83	48.04	62.83	52.7
falling	33.52	91.53	67.39	50.32	56.11
jumping	31.57	94.86	56.07	51.86	53.4
l-pass	51.8	80.14	37.9	40.77	52.34
1-spike	55.27	95.56	53.08	63.24	63.97
1_set	55.36	96.11	40.51	34.66	56.69
1_winpoint	59.97	98.19	52.9	71.02	67.51
moving	59.96	81.81	43.34	97.31	65.78
r-pass	53.99	82.22	68.37	58.97	63.98
r_set	52.59	77.22	56.2	61.56	59.77
r_spike	56.54	20	57.18	84.83	53.64
r_winpoint	48.89	86.81	46.19	64.1	58.08
setting	56.19	90.56	45.68	72.79	62.93
spiking	57.22	56.81	33.42	48.86	49.93
standing	36.97	94.58	25.16	73.03	50.95
waiting	47.11	91.53	96.2	40.52	67.28

Table 5. Ablation Study of Model Components.

Model Components	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
YOLOv8 Detection Only	85.2	83.7	84.4	84.0
DeepSORT Tracking Only	88.1	86.3	87.0	86.6
CNN-only Classification	90.5	89.1	89.7	89.4
CNN-LSTM Full Model	94.3	92.8	93.1	92.9

the complete CNN–LSTM model. The table shows that the YOLOv8 detection module alone achieves an accuracy of 85.2%, while DeepSORT tracking improves the performance to 88.1%. The CNN-only classification model further enhances accuracy to 90.5%. Finally, the complete CNN–LSTM framework demonstrates the highest performance, achieving an accuracy of 94.3%, highlighting the effectiveness of combining detection, tracking, and temporal modeling for improved action recognition.

4. Conclusion

The suggested deep learning-driven volleyball teaching system effectively brings together YOLOv8 detection, DeepSORT tracking, and a hybrid CNN-LSTM architecture to provide precise and automated analysis of player actions. The model not only identifies the players but also tracks their movements and classifies crucial actions like spike, serve, and block with very high reliability by the method of efficiently processing volleyball match videos. Data preprocessing and augmentation performance has also been reflected in model stability and generalization. Skill assessment done by predicting motion patterns gives meaningful feedback which is beneficial for both players' learning and coaches' decision-making. The system not only offers a practical solution for modern sports analytics but also legitimizes the role of AI in making athletic training more efficient. In the future, one of the main aims of the project might be to incorporate more diverse match situations into the dataset in order to make the model more adaptable to different types of court settings. Moreover, including 3D pose estimation could help in the analysis of players' movements and give more precise skill evaluation. The system could be made available in real-time on edge computer devices or mobile platforms like smartphones during on-the-spot training sessions. Also, the addition of a feature to give training advice based on individual needs could facilitate the gradual growth of the player.

Declaration

Data Availability

https://www.kaggle.com/datasets/sherif31/group-activity-recognition-volleyball?utm_source=chatgpt.com

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Funding Statement

This research received no external funding.

Author contribution

Ruibi Chen:

Conceptualization, methodology, data curation, software, writing—original draft.

Shangfu Meng:

Investigation, validation, visualization, writing—review and editing.

Wei Sun:

upervision, project administration, resources, final approval of the manuscript.

Ethical approval

This article does not involve any studies with human participants or animals performed by any of the authors. All procedures followed ethical research standards.

Consent to participate

Not applicable, as no human participants were involved in the study.

Consent to publication

All authors have read and approved the final manuscript and consent to its publication.

Competing interests

The authors declare that they have no competing interests.

Acknowledgment

The authors would like to thank Wenzhou University of Technology, Beijing Vocational College of Labour and Social Security, and Beijing City University for their support in facilitating this research.

References

[1] Y. Zhang, W. Duan, L. E. Villanueva, and S. Chen, (2023) *"Transforming sports training through the integration of internet technology and artificial intelligence"* **Soft Computing 27**(20): 15409–15423. DOI: 10.1007/s00500-023-08960-w.

[2] G. Yuan, (2024) *"Application of posture estimation optimization algorithm in the analysis of college air volleyball teaching movements"* **Systems and Soft Computing *6***: 200135. DOI: 10.1016/j.sasc.2024.200135.

[3] N. Kucirkova, L. Gerard, and M. C. Linn, (2021) *"Designing personalised instruction: A research and design framework"* **British Journal of Educational Technology *52***(5): 1839–1861. DOI: 10.1111/bjet.13119.

[4] Y. Jia et al., (2025) *"A narrative review of deep learning applications in sports performance analysis: current practices, challenges, and future directions"* **BMC Sports Science, Medicine and Rehabilitation *17***(1): 249. DOI: 10.1186/s13102-025-01294-0.

[5] P. Rule, (2024) *"Dialogue, Horizon and Chronotope: Using Bakhtin's and Gadamer's Ideas to Frame Online Teaching and Learning"* **Studies in Philosophy and Education *43***(3): 305–323. DOI: 10.1007/s11217-024-09933-8.

[6] T. Zhang, C. Jiao, H. Sun, and X. Liang, (2022) *"Application of Internet of Things Combined with Wireless Network Technology in Volleyball Teaching and Training"* **Computational Intelligence and Neuroscience *2022***: 8840227. DOI: 10.1155/2022/8840227.

[7] R. Bucea-Manea-Țoniș, L. Vasile, R. Stănescu, and A. Moanță, (2022) *"Creating IoT-Enriched Learner-Centered Environments in Sports Science Higher Education during the Pandemic"* **Sustainability *14***(7): 4339. DOI: 10.3390/su14074339.

[8] H. Yu and Y. Mi, (2023) *"Application Model for Innovative Sports Practice Teaching in Colleges Using Internet of Things and Artificial Intelligence"* **Electronics *12***(4): 874. DOI: 10.3390/electronics12040874.

[9] J. Shen and L. Chen, (2024) *"Application of Human Posture Recognition and Classification in Performing Arts Education"* **IEEE Access *12***: 125906–125919. DOI: 10.1109/ACCESS.2024.3451172.

[10] J. Bridgeman and A. Giraldez-Hayes, (2024) *"Using artificial intelligence-enhanced video feedback for reflective practice in coach development: benefits and potential drawbacks"* **Coaching: An International Journal of Theory, Research and Practice *17***(1): 32–49. DOI: 10.1080/17521882.2023.2228416.

[11] R. Baranyi, Y. Körber, P. Galimov, Z. Parandeh, and T. Grechenig, (2023) *"Rehafox – A therapeutical approach developing a serious game to support rehabilitation of stroke patients using a leap motion controller"* **Clinical eHealth *6***: 85–95. DOI: 10.1016/j.ceh.2023.08.001.

[12] W. Mao, (2022) *"Video analysis of intelligent teaching based on machine learning and virtual reality technology"* **Neural Computing and Applications *34***(9): 6603–6614. DOI: 10.1007/s00521-021-06072-w.

[13] F. Xiang, J. Cao, Y. Zuo, X. Duan, L. Xie, and M. Zhou, (2024) *"A Novel Training Path to Promote the Ability of Mechanical Engineering Graduates to Practice and Innovate Using New Information Technologies"* **Sustainability *16***(1): 364. DOI: 10.3390/su16010364.

[14] F. Cao, M. Xiang, K. Chen, and M. Lei, (2022) *"Intelligent Physical Education Teaching Tracking System Based on Multimedia Data Analysis and Artificial Intelligence"* **Mobile Information Systems *2022***: 7666615. DOI: 10.1155/2022/7666615.

[15] J. Feng, (2023) *"Designing an Artificial Intelligence-based sport management system using big data"* **Soft Computing *27***(21): 16331–16352. DOI: 10 . 1007 / s00500-023-09162-0.

[16] C. Duan, (2021) *"Design of online volleyball remote teaching system based on AR technology"* **Alexandria Engineering Journal *60***(5): 4299–4306. DOI: 10.1016/j.aej.2021.03.006.

[17] D. C. Mănescu, (2025) *"Big Data Analytics Framework for Decision-Making in Sports Performance Optimization"* **Data *10***(7): 116. DOI: 10.3390/data10070116.

[18] Z. Hu, Z. Liu, and Y. Su, (2024) *"AI-Driven Smart Transformation in Physical Education: Current Trends and Future Research Directions"* **Applied Sciences *14***(22): 10616. DOI: 10.3390/app142210616.

[19] P. Liu and E. Dastbaravardeh, (2025) *"Deep Learning-Driven Assessment of Student Movement and Performance Using Physiological Data in Physical Education Information Systems: An S-AIoT Solution"* **International Journal of Intelligent Systems *2025***: 9479311. DOI: 10.1155/int/9479311.

[20] P. Pietraszewski et al., (2025) *"The Role of Artificial Intelligence in Sports Analytics: A Systematic Review and Meta-Analysis of Performance Trends"* **Applied Sciences *15***(13): 7254. DOI: 10.3390/app15137254.

[21] S. Caso, P. Furley, and G. Jordet, (2025) *"Using video-notational analysis to examine soccer players' behaviours"* **International Journal of Sport and Exercise Psychology**: 1–21. DOI: 10.1080/1612197X.2025.2477165.

[22] M. Batez, T. Petrušič, Š. Bogataj, and N. Trajković, (2021) *"Effects of Teaching Program Based on Teaching Games for Understanding Model on Volleyball Skills and Enjoyment in Secondary School Students"* **Sustainability *13***(2): 606. DOI: 10.3390/su13020606.

[23] F. Sgrò, M. Barca, R. Schembri, R. Coppola, and M. Lipoma, (2022) *"Effects of different teaching strategies on students' psychomotor learning outcomes during volleyball lessons"* **Sport Sciences for Health *18***(2): 579–587. DOI: 10.1007/s11332-021-00850-8.

[24] D. Stojanović et al., (2023) *"School-Based TGfU Volleyball Intervention Improves Physical Fitness and Body Composition in Primary School Students: A Cluster-Randomized Trial"* **Healthcare *11***(11): 1600. DOI: 10.3390/healthcare11111600.

[25] X. Dai and S. Li, (2021) *"Volleyball Data Analysis System and Method Based on Machine Learning"* **Wireless Communications and Mobile Computing *2021***: 9943067. DOI: 10.1155/2021/9943067.

[26] S. Leng and M. Shao, (2022) *"A Study on the Effect of the Club Model on the Effectiveness of College Volleyball Teaching Based on a Random Matrix Model"* **Mathematical Problems in Engineering *2022***: 5681412. DOI: 10.1155/2022/5681412.

[27] H. Wu, (2021) *"Evaluation of AdaBoost's elastic net-type regularized multi-core learning algorithm in volleyball teaching actions"* **Wireless Networks**: DOI: 10.1007/s11276-021-02694-z.

[28] F. A. Salim, D. B. W. Postma, F. Haider, S. Luz, B.-J. F. van Beijnum, and D. Reidsma, (2024) *"Enhancing volleyball training: empowering athletes and coaches through advanced sensing and analysis"* **Frontiers in Sports and Active Living *6***: DOI: 10.3389/fspor.2024.1326807.

[29] W. Jiang, K. Zhao, and X. Jin, (2021) *"Diagnosis Model of Volleyball Skills and Tactics Based on Artificial Neural Network"* **Mobile Information Systems *2021***: 7908897. DOI: 10.1155/2021/7908897.

[30] B. Liu, N. Yang, X. Han, and C. Liu, (2021) *"Neural Network for Intelligent and Efficient Volleyball Passing Training"* **Mobile Information Systems *2021***: 3577541. DOI: 10.1155/2021/3577541.

[31] A. Ferriz-Valero, O. Østerlie, S. García-Martínez, and S. Baena-Morales, (2022) *"Flipped Classroom: A Good Way for Lower Secondary Physical Education Students to Learn Volleyball"* **Education Sciences *12***(1): 26. DOI: 10.3390/educsci12010026.

[32] Z. Zhang, (2021) *"Analysis of Volleyball Video Intelligent Description Technology Based on Computer Memory Network and Attention Mechanism"* **Computational Intelligence and Neuroscience *2021***: 7976888. DOI: 10.1155/2021/7976888.

[33] R. Schweighardt, (2023) *"Flipping the Script on Teaching Volleyball"* **Strategies *36***(3): 3–7. DOI: 10.1080/08924562.2023.2195210.

[34] L. Jiang, Z. Yang, and L. Gang, (2025) *"Transformer-Based Multi-Player Tracking and Skill Recognition Framework for Volleyball Analytics"* **IEEE Access *13***: 8806–8824. DOI: 10.1109/ACCESS.2025.3526775.

[35] J. B. Apidogo, A. Ammar, A. Salem, J. Burdack, and W. I. Schöllhorn, (2024) *"Resonance Effects in Variable Practice for Handball, Basketball, and Volleyball Skills: A Study on Contextual Interference and Differential Learning"* **Sports *12***(1): 5. DOI: 10.3390/sports12010005.

[36] S. McCormack, B. Jones, D. Elliott, D. Rotheram, and K. Till, (2022) *"Coaches' Assessment of Players Physical Performance: Subjective and Objective Measures are needed when Profiling Players"* **European Journal of Sport Science *22***(8): 1177–1187. DOI: 10.1080/17461391.2021.1956600.