ITC 3/55
Information Technology and Control
Vol. 55 / No. 3/ 2026
pp. 1425-1443
DOI 10.5755/j01.itc.55.3.45015

Badminton Swing Analysis Using Lightweight OpenPose with an Inception-Time Module for Temporal Modeling

Received 2026/05/13 | Accepted after revision 2026/08/05

**HOW TO CITE:** Huang, J., Xie, Y., Deng, X., Li, Y., Ye, M. (2026) Badminton Swing Analysis Using Lightweight OpenPose with an Inception-Time Module for Temporal Modeling. *Information Technology and Control*, 55(3), 1424-1442. https://doi.org/10.5755/j01.itc.55.3.45015

# Badminton Swing Analysis Using Lightweight OpenPose with an Inception-Time Module for Temporal Modeling

**Jia Huang, Yu Xie***

School of Physical Education, Hunan University, Changsha, 410082, China; e-mails: 18932462566@139.com, anning20042217@126.com

**Xiaolei Deng**

Department of Physical Education, Changsha Vocational and Technical College, Changsha, 410217, China; e-mail: 18675142505@139.com

**Yingqing Li, Mengjie Ye**

School of Physical Education, Hunan University, Changsha, 410082, China; e-mails: libao21997@163.com, xieyupe@hnu.edu.cno

**Corresponding author:** anning20042217@126.com

In badminton training and technical and tactical analysis, accurately identifying and standardly evaluating racket swings are of great significance for improving athlete performance and preventing sports injuries. To analyze swing movements. This paper presents an integrated pipeline for badminton swing analysis, combining a lightweight OpenPose variant (with ShuffleNet-V2 backbone, adaptive resolution, and Kalman filtering) with an Inception-based architecture augmented by a temporal convolution branch (termed Inception-Time). The temporal branch specifically captures short-term phase transitions in swings, and a dual-branch output jointly optimizes action classification and normative scoring. The action recognition accuracy of LOP-NGN on the self-built BSD data set and the public SPB data set reached 96.5% and 97.3%, respectively, and the single frame response time was 28ms and 30ms, respectively. In the actual swing analysis scenario, the average absolute errors in elbow, shoulder, and wrist joint angle estimation were 1.2°, 1.4°,

and 1.6° under low-speed swing conditions, and 1.5°, 1.8°, and 2.1° under high-speed swing conditions. The system power consumption was approximately 18.5 to 19.3W, the analysis frame rate reached 30~32 FPS, and the scoring error was 4.2~4.5 points. The results show that the method can efficiently and accurately identify and score badminton swings, significantly reducing computational overhead and joint estimation errors, and providing reliable technical support for intelligent badminton training.

KEYWORDS: Badminton swing analysis; Lightweight OpenPose; New GoogleNet; Action recognition; Normative scoring.

## 1. Introduction

Badminton is a sport that requires extremely high technical precision. The standardization of its swing directly affects the quality of the shot and the risk of sports injuries. Driven by computer vision and deep learning, action analysis systems based on human posture estimation provide objective and quantitative evaluation methods for sports training [19, 25]. Currently, research on badminton swing motion analysis mainly focuses on single motion recognition or simple biomechanical parameter extraction. It lacks temporal modeling and multi-dimensional evaluation of the entire swing process, resulting in challenges still faced in fine-grained identification and normative scoring of complex swing motions (such as smashes, drop shots, and long shots) [2, 30]. Existing methods mostly use general pose estimation models (such as OpenPose, MediaPipe) combined with LSTM or TCN for action classification. However, there are common problems such as large number of model parameters, poor real-time performance, obvious jitter of key points, and difficulty in balancing classification and scoring tasks [27]. The research develops a badminton swing analysis method based on lightweight OpenPose and New GoogleNet. First, ShuffleNet-V2 is used to replace the original backbone network of OpenPose, and combined with adaptive resolution adjustment and Kalman filter temporal smoothing. It significantly reduces the amount of calculation and suppresses joint trajectory jitter while ensuring detection accuracy. Secondly, the Inception-Time module is designed and the temporal convolution branch is introduced into New GoogleNet to enhance the ability to capture phase changes in the swing movement. A dual-branch output structure is constructed to jointly optimize action category recognition and normative scoring. The contribution is an engineering integration that combines a lightweight pose estimator (ShuffleNet-V2 + adaptive resolution + Kalman smoother) with a temporal-aware Inception module (with a 3×1 convolution branch) in a dual-branch framework. This combination is motivated by the need for both spatial precision and temporal continuity in swing analysis; it achieves practical gains in inference speed and joint-angle stability on the collected dataset, without claiming novel operators or training paradigms.

## 2. Related Work

New GoogleNet and lightweight OpenPose are a combination of deep learning models used in the field of human motion analysis. It combines lightweight posture estimation and efficient convolutional neural network technology. It can extract key points of human skeletons and identify complex action patterns in real time, and realize a series of analysis tasks such as detection, classification, and normative scoring of badminton swings. It is crucial in modern intelligent sports training and action evaluation systems. A large number of scholars have conducted relevant research on it. Lu et al. [18] built a dual-pass method integrating EfficientNetV2-S and OpenPose to address the lack of action recognition information in a single static image. The residual spatial attention enhancement feature was embedded in the main channel, and one-dimensional convolution was used to adaptively fuse the classification results to achieve superior accuracy on the Stanford 40 data set. [18]. Aiming at the difficulty of three-dimensional pose estimation from low-resolution infrared sensors, Chen et al. proposed a skeleton estimation system with attention mechanism and CNN-BiGRU. They used 8×8

infrared array acquisition, Kinect v2 annotation, and k-means to eliminate outliers, achieving 90.151% accurate restoration within 20mm [4]. Lu et al. [17] proposed a TCNet hybrid network for the low efficiency of long-range spatiotemporal interaction capture in continuous sign language recognition. The trajectory module extracted regional motion patterns, and the relevant area module dynamically filtered out irrelevant frames, achieving leading word error rate reduction on the PHOENIX14 data set [17]. Shah et al. [24] systematically reviewed the evolution and limitations of mainstream frameworks from YOLO to Cascade R-CNN. An improved framework that integrates open source libraries was proposed to enhance real-time dimensional change tracking and measurement, which could effectively meet the diverse needs of industrial vision [24]. Aiming at the insufficient utilization of human body structure information in wearable human activity recognition, Zou et al. [29] proposed a graph-based temporal framework GT-WHAR. A human perceptual graph was designed to uniformly represent the body motion structure. The body node attention graph network was used to fuse cross-channel information. A bidirectional temporal network was embedded to capture features, which was significantly better than existing methods [29]. Ibh et al. [13] proposed TemPose, a skeleton-based Transformer model with temporal and interaction layers for fine-grained badminton action recognition, which integrated shuttlecock and player position data and outperformed baseline models on badminton datasets [13].

Since the development of badminton action recognition, some of its theories and practical applications have become relatively mature. Scholars have conducted in-depth research. In view of the unclear transmission mechanism of the joint power chain in the badminton smashing action and the insufficient quantification of factors affecting batting efficiency, Guo et al. [10] conducted a kinematic analysis of the in-situ smashing techniques of 40 national-level athletes. A motion capture system was used to collect three-dimensional coordinate trajectories, calculate the peak angular velocity and delay of the knee, hip, shoulder, elbow, and wrist joints, and analyze the power chain transmission sequence and contribution distribution. The forward flexion of the shoulder joint contributed the most, excessive speed of the elbow joint hindered speed transmission, and the position of the center of gravity was significantly related to the acceleration of the clap [10]. Edmizal et al. [8] conducted a systematic review on the biomechanical differences between elites and amateurs in badminton smash action. Multiple relevant databases were searched, literature was screened according to inclusion and exclusion criteria, and biomechanical variables such as movement patterns, beat speed, and coordination were extracted for comparative analysis. Analysis showed that elite athletes had better action patterns and beat speeds, and related features could be used as training targets [8]. Artazila et al. [1] used the PRISMA standard to review the literature in the Scopus database from 2020 to 2024 in response to the lack of systematic integration of biomechanical analysis of badminton smash technology. The badminton and smash keywords were used to search, and 10 relevant articles were selected from 155 articles for comprehensive analysis. Analysis results showed that a biomechanical perspective could improve smash performance, speed, and strength training effects [1]. Purnama et al. [21] proposed a time series classification method that combined landmark tracking and LSTM to capture motion trajectory features in badminton posture time series classification. Key point trajectories were extracted through posture estimation. The proposed method could identify badminton action posture categories [21]. Larasaty et al. [14] used IMU sensors to analyze the kinematics of the wrist and forearm in different service types to solve the insufficient quantification of upper limb joint angle motion characteristics in badminton service technology. Sensors were placed on the back of the hand and forearm near the elbow to measure the angular movement patterns of backhand, short forehand, and long forehand serves. Forehand serving was mainly driven by the forearm, while backhand serving had a more significant contribution from the wrist [14]. There is a lack of theoretical model for optimizing the shortest hitting time trajectory in badminton smash. Putra et al. [22] introduced the fastest drop line into the smash kinematics analysis. The smash trajectory under different conditions was simulated through analysis and numerical simulation. The cycloid trajectory could achieve the shortest hitting time, and the ratio of smash to high and long ball speed was 0.75. Experimental results showed that the model was consistent with literature data [22].

To sum up, although many scholars have conducted research on human posture estimation and action recognition technology, there is no method that combines lightweight OpenPose with New GoogleNet and applies it to real-time analysis and normative scoring of badminton swing movements. Therefore, the study proposes a badminton swing analysis method based on lightweight OpenPose and New GoogleNet to reduce joint angle estimation errors, and improve action recognition accuracy and scoring stability, providing theoretical support for intelligent badminton training and action technology diagnosis.

## 3. Methods

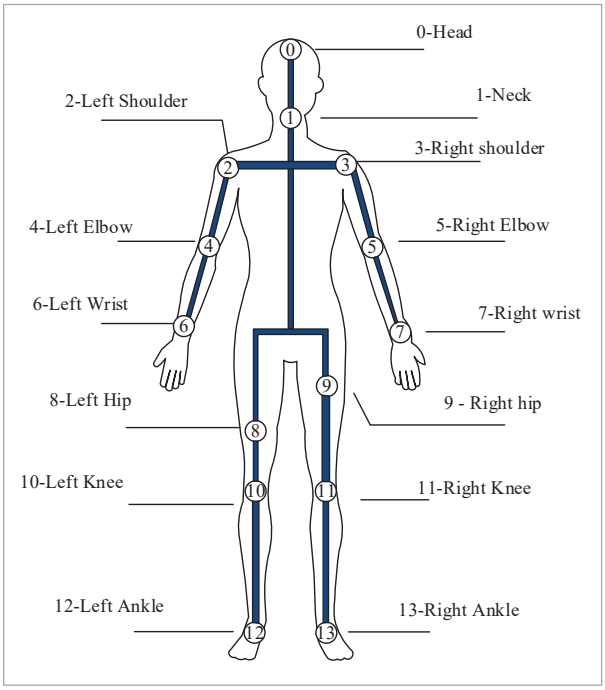
In view of the high complexity and strong real-time requirements of badminton swing, a two-stage cascaded analysis pipeline is designed. This work consists of two core modules connected in series. First, the lightweight OpenPose pose estimation module is responsible for extracting key human skeleton points from the video stream in real time. Then, the New GoogleNet action recognition and evaluation module performs action classification and normative scoring based on time-series skeletal features. The two modules work together to achieve accurate analysis and quantitative evaluation of swing movements. It is worth clarifying that these two modules are trained separately rather than jointly; the pose estimator is pre-trained and then fixed during training of the action recognition module, so the overall pipeline is not end-to-end in the strict sense.

### 3.1. Lightweight OpenPose Pose Estimation Module

#### 3.1.1. Lightweight Backbone Network and Adaptive Temporal Smoothing

The lightweight OpenPose pose estimation process usually consists of multiple links, including skeleton key point topology definition, vector field detection (Part Affinity Fields, PAF), backbone network feature extraction based on ShuffleNet-V2, and post-processing filtering and smoothing, etc. [5]. In these links, the accurate definition of skeletal key points and limb association play a vital role to ensure that subsequent action analysis can be based on a stable and continuous human skeleton sequence [16]. The estimation framework involves the collaborative operation of multiple subtasks, including multi-scale feature extraction, parallel branch prediction, and temporal smoothing. When analyzing badminton swing movements, all aspects such as real-time detection of player posture, reliable estimation of key point confidence, and jitter suppression of joint trajectories is performed. The lightweight OpenPose is to reduce the amount of model calculations and parameters while ensuring detection accuracy, thereby achieving real-time pose estimation on edge devices. Its key process is shown in Figure 1.

**Figure 1**
Definition of skeletal key points for badminton racket swing motion.




The 14-keypoint topology is defined based on biomechanical relevance to swing kinematics: torso centers (shoulder/hip) encode trunk rotation and weight transfer, while limb endpoints (wrist/ankle) track the distal power-chain (shoulder-elbow-wrist) critical for differentiating smashes from clears. This subset deliberately excludes facial and hand-finger joints (27 in full OpenPose) to reduce computational cost, while retaining sufficient degrees of freedom to capture the three swing phases (wind-up, impact, and follow-through). Figure 1 shows the specific layout of these 14 key points. The core parts of the torso (such as the shoulder and hip centers) are used to capture

the main body rotation and changes in the center of gravity. The ends of the limbs (such as wrists and ankles) are used to accurately depict the acceleration, hitting and follow-up movements of the arms during the swing. The set of key points detected in frame $t$ is $K_t=\{\mathbf{k}_{t,i} \mid i=1,...,14\}$, where $\mathbf{k}_{t,i}=(x_{t,i}, y_{t,i}, c_{t,i})$ and $x_{t,i}$, $y_{t,i}$ are pixel coordinates, and $c_{t,i}$ is confidence. All key point coordinates are normalized to the [-1,1] interval before being input into the network, as shown in Equation (1) [28].

$$\hat{x}_{t,i}=\frac{2x_{t,i}}{W}-1,\quad \hat{y}_{t,i}=\frac{2y_{t,i}}{H}-1. \tag{1}$$

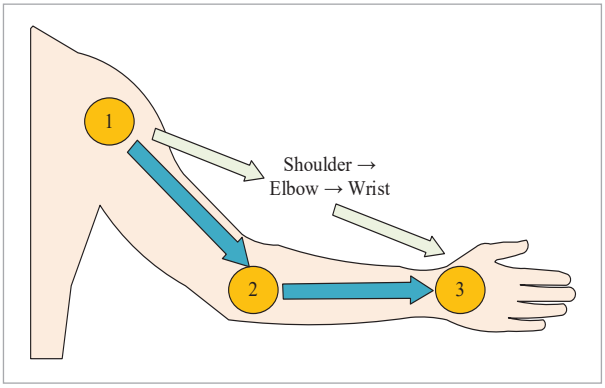
In Equation (1), $x_{t,i}$ and $y_{t,i}$ respectively represent the horizontal and vertical coordinates of the key points in the original pixel coordinate system. $\hat{x}_{t,i}$ and $\hat{y}_{t,i}$ respectively represent the normalized horizontal and vertical coordinates of key points. $W$ and $H$ represent the image width and height. Lightweight OpenPose uses a bottom-up detection strategy to simultaneously predict key point confidence maps and PAF. For the $j$-th key point category, its confidence map $\mathbf{S}_j$ is defined as Gaussian response superposition, as shown in Equation (2) [15].

$$\mathbf{S}_j(\mathbf{p})=\max_m \exp\left(-\frac{\| \mathbf{p}-\mathbf{k}_{t,j}^{(m)}\|_2^2}{\sigma^2}\right). \tag{2}$$

In Equation (2), $\mathbf{S}_j(\mathbf{p})$ represents the value of the confidence map of the $j$-th key point at pixel $\mathbf{p}$. $\mathbf{p}$ represents any pixel position in the image. $\mathbf{k}_{t,j}^{(m)}$ represents the true coordinates of the $j$ of the $m$-th person. $\sigma$ represents the Gaussian kernel width. PAF describes the limb connection direction between two key points. The following is a schematic diagram of PAF encoding the right arm (shoulder-elbow-wrist), as shown in Figure 2.

In Figure 2, the way the arrows are superimposed on the right arm area of the human body indicates the PAF direction encoding from the right shoulder to the right elbow, and then from the right elbow to the right wrist. The dense area of arrows signifies the direction of the trunk of the limb, with the direction from the shoulder to the wrist, and the depth of the color indicates the confidence of the vector. For the limb connecting key points $j_1$ and $j_2$, its vector field

**Figure 2**
Schematic diagram of the right arm PAF vector field.




$\mathbf{L}_c$ is defined as follows. The pixel $\mathbf{p}$ is located in the strip area of the limb, as shown in Equation (3) [11].

$$\mathbf{L}_c(\mathbf{p})=\frac{\mathbf{k}_{t,j_2}-\mathbf{k}_{t,j_1}}{\| \mathbf{k}_{t,j_2}-\mathbf{k}_{t,j_1}\|_2}. \tag{3}$$

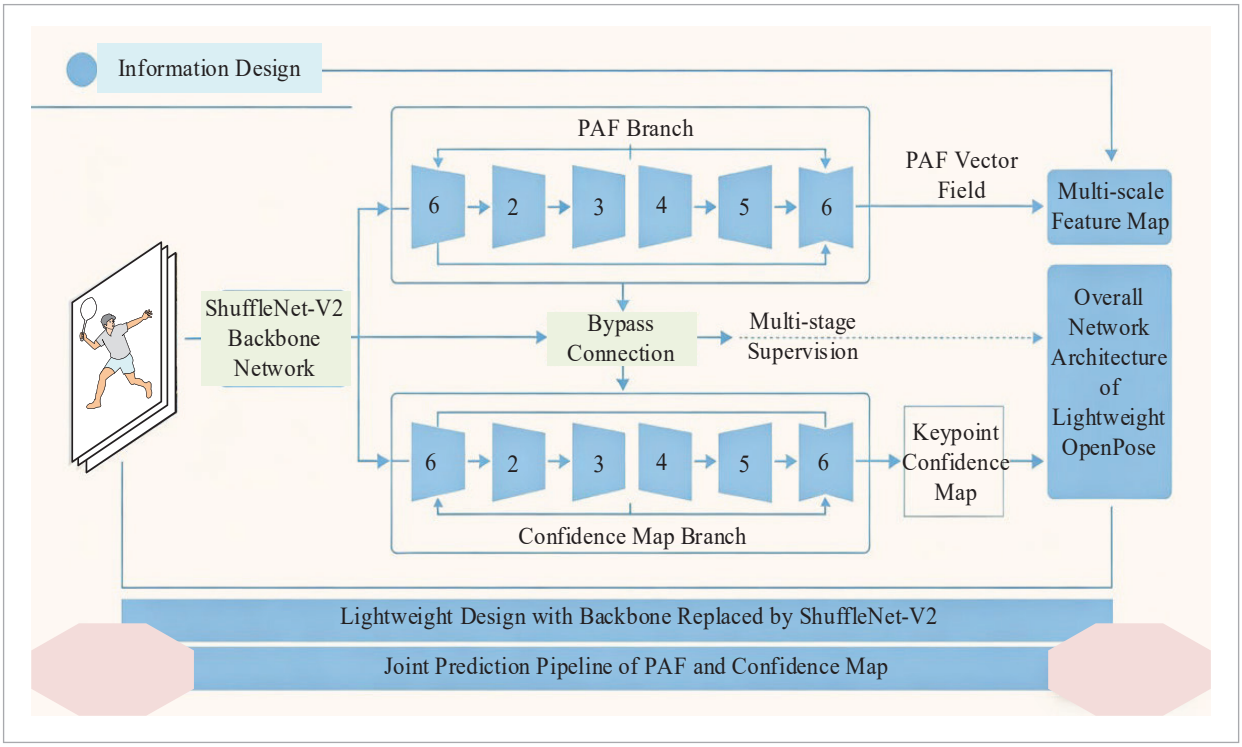
In Equation (3), $\mathbf{L}_c(\mathbf{p})$ signifies the value of the PAF vector of the $c$-th limb at pixel $\mathbf{p}$. $\mathbf{k}_{t,j_2}$ and $\mathbf{k}_{t,j_1}$ respectively represent the real coordinates of the key points at both ends of the limb.

### 3.1.2. Lightweight backbone network and post-processing optimization

The original OpenPose uses VGG-19 as the backbone network, which has many parameters and high computational overhead, making it difficult to achieve real-time inference on edge devices [23]. To satisfy the strict real-time requirements of badminton swing analysis, the study replaces it with the lightweight ShuffleNet-V2. Through channel segmentation and channel shuffling operations, this network greatly reduces the amount of calculation while maintaining high detection accuracy, and significantly improves the inference efficiency of the model [3]. The overall network structure diagram of lightweight OpenPose is shown in Figure 3.

Figure 3 shows the detailed architecture of the lightweight OpenPose network. The input passes through ShuffleNet-V2 backbone, followed by a 6-stage cascade. Each stage contains two parallel branches: PAF (limb association) and confidence map (joint localization). Feature map dimensions are annotated at each stage (e.g., 64×64×128). The

**Figure 3**
Lightweight OpenPose network architecture




final outputs are aggregated for pose reconstruction. The latter stage takes the output and backbone features of the previous stage as input to gradually refine the prediction results, thus improving the robustness of limb association and key point positioning accuracy under occlusion. The calculation amount (FLOPs) of a ShuffleNet-V2 basic unit is shown in Equation (4) [6].

$$\text{FLOPs}_{\text{unit}} = \frac{1}{2}\cdot C_{\text{in}}\cdot H\cdot W\cdot K^2\cdot\frac{C_{\text{out}}}{2}+\frac{1}{2}\cdot C_{\text{in}}\cdot H\cdot W\cdot K^2\cdot\frac{C_{\text{out}}}{2}. \tag{4}$$

In Equation (4), $\text{FLOPs}_{\text{unit}}$ signifies the calculation amount of the ShuffleNet-V2 basic unit. $C_{\text{in}}$ and $C_{\text{out}}$ signify the number of input channels and the number of output channels. $K$ signifies the convolution kernel size. Compared with the standard residual unit, the calculation amount is reduced by about 40%. The loss function of the network has three parts, key point confidence loss, PAF loss, and auxiliary intermediate supervision loss. The total loss is shown in Equation (5) [7].

$$L_{\text{total}} = \sum_{s=1}^{S}\left(L_{\text{conf}}^{(s)}+L_{\text{paf}}^{(s)}\right)+L_{\text{mid}}. \tag{5}$$

In Equation (5), $L_{\text{total}}$ signifies the total loss of the network. $S$ represents the number of prediction stages. $L_{\text{conf}}^{(s)}$ signifies the key point confidence loss in the $s$. $L_{\text{paf}}^{(s)}$ signifies the PAF loss in the $s$-th stage. $L_{\text{mid}}$ signifies the auxiliary intermediate supervision loss. $S$=6 is the number of prediction stages. The confidence loss at each stage uses weighted binary cross-entropy, as shown in Equation (6) [12].

$$L_{\text{conf}}^{(s)} = -\frac{1}{N}\sum_{j=1}^{J}\sum_{\mathbf{p}}W(\mathbf{p})\cdot\left[\mathbf{S}_j^{\text{gt}}(\mathbf{p})\log\mathbf{S}_j^{(s)}(\mathbf{p})+(1-\mathbf{S}_j^{\text{gt}}(\mathbf{p}))\log(1-\mathbf{S}_j^{(s)}(\mathbf{p}))\right]. \tag{6}$$

In Equation (6), $N$ signifies the normalization constant. $J$ signifies the number of key point categories.

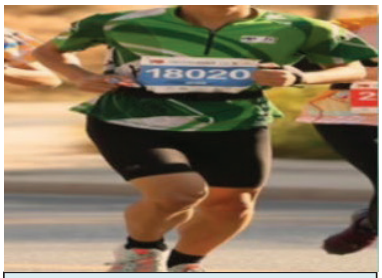
$W(\mathbf{p})$ represents the mask function. $S_j^{gt}(\mathbf{p})$ signifies the true confidence map of the $j$-th key point. $S_j^{(s)}(\mathbf{p})$ represents the confidence map of the $j$-th type key points in the $s$-th stage. The PAF loss form is similar, except waathat $S$ is replaced by $L$. To detect small limbs of long-distance athletes, a dynamic resolution adjustment strategy is introduced. According to the height of the human body bounding box detected in the previous frame, the input resolution of the current frame is determined according to the following rules, as shown in Equation (7) [20].

$$R = \min\left(368, \max\left(184, \left\lfloor \frac{368 \times 184}{h_{box}} \right\rfloor\right)\right). \quad (7)$$

In Equation (7), $R$ represents the input resolution after dynamic adjustment. $h_{box}$ signifies the height of the human bounding box detected in the previous frame. and represent the minimum and maximum function. To quantitatively evaluate the impact of different input resolution strategies on detection accuracy and efficiency, the study compares three solutions, fixed high resolution (368×368), fixed low resolution (184×184), and the adaptive resolution proposed in the study (dynamically adjusted according to the height of the human body bounding box). Figure 4 illustrates the processing effect of the adaptive resolution strategy on a single-frame swing image.

In Figure 4, the adaptive resolution strategy dynamically determines the input resolution of the current frame based on the height of the human body bounding box detected in the previous frame. When the athlete is farther from the camera and the height of the bounding box is smaller, a higher resolution is used to preserve small body details such as wrists and elbows. When the athlete is closer to the camera and the height of the bounding box is larger, the resolution is appropriately reduced to control the computational load. This strategy achieves an adaptive balance between detection accuracy and inference efficiency without relying on a fixed resolution threshold. Compared with the fixed high resolution, the adaptive strategy significantly improves the inference efficiency with a slight decrease in key point accuracy. Compared with the fixed low resolution, the adaptive strate-

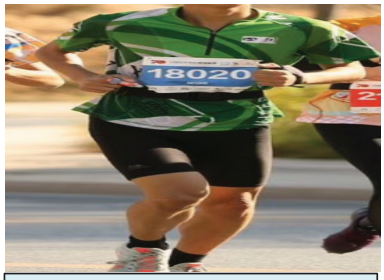
**Figure 4**
Schematic diagram of adaptive resolution input (Images were captured by the authors at a local badminton training center using a Basler acA2440-75um camera (75 fps, 1920×1080 resolution) under indoor lighting conditions.).




Fixed 368 Resolution:

Rich details, clear small limbs, high computational cost

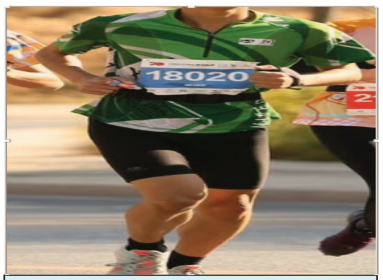
Fixed 368 Resolution




Fixed 184 Resolution:

Blurred image, missing detection/ positioning deviation of small limbs, fast inference speed

Fixed 184 Resolution




Adaptive Resolution:

Dynamically adjust the size, retain details, and avoid computational redundancy

Adaptive Resolution

gy greatly improves the detection accuracy while sacrificing a small amount of speed. Since the rapid swing motion causes high-frequency jitter in the trajectory of key points, the study uses Kalman filtering to smooth the time series of the coordinates of each key point. The state vector is defined. The state transition is shown in Equation (8) [9].

$$\mathbf{z}_{t,i} = \mathbf{F}\mathbf{z}_{t-1,i} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathrm{N}(0, \mathbf{Q}). \tag{8}$$

In Equation (8), $\mathbf{z}_{t,i}$ signifies the state vector of the $i$-th key point at time $t$. $\mathbf{F}$ represents the state transition matrix. $\mathbf{w}_t$ represents process noise, which obeys a Gaussian distribution with mean 0 and covariance $\mathbf{Q}$. $\mathbf{Q}$ represents the process noise covariance matrix. The observation model is shown in Equation (9) [26].

$$\mathbf{o}_{t,i} = \mathbf{H}\mathbf{z}_{t,i} + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathrm{N}(0, \mathbf{R}). \tag{9}$$

In Equation (9), $\mathbf{o}_{t,i}$ signifies the observation vector of the $i$-th key point at time $t$. $\mathbf{H}$ represents the observation matrix. $\mathbf{v}_t$ represents the observation noise, which obeys the Gaussian distribution with mean 0 and covariance $\mathbf{R}$. $\mathbf{R}$ represents the observation noise covariance matrix. The process noise covariance Q and observation noise covariance R are not set heuristically. They are tuned via grid search on a validation subset to minimize the RMSE of tracked joint positions against high-speed camera annotations (ground truth). Specifically, Q is scaled proportionally to the expected squared acceleration of each joint (higher for wrist/ankle than for torso centers), while R is set inversely proportional to the detection confidence score from the PAF branch, enabling adaptive per-keypoint filtering. This configuration reduces trajectory jitter by ~40% compared to a fixed-gain low-pass filter in preliminary experiments.

## 3.2. New Googlenet Action Recognition and Evaluation Module

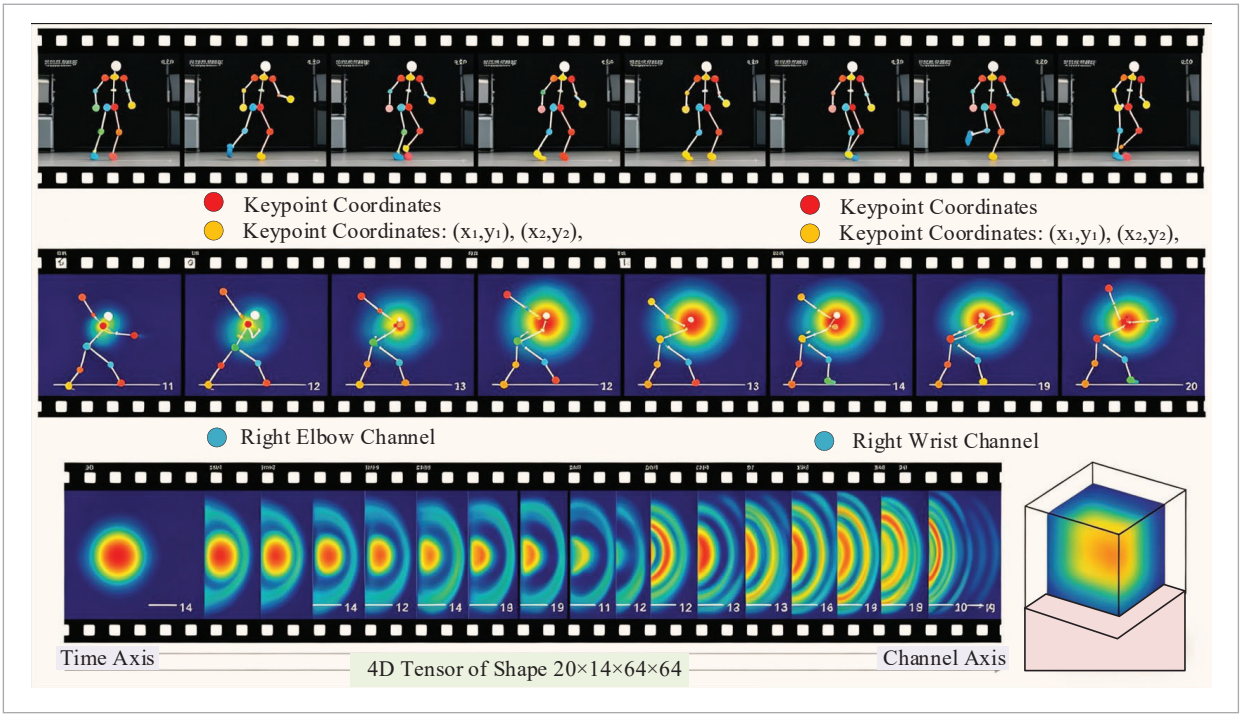
### 3.2.1. Spatiotemporal Heat Map Encoding of Skeleton Point Sequence

The module receives a sequence of 20 consecutive frames of skeleton points generated by the above-mentioned research. To determine the window size T, the swing-duration distribution in the dataset (captured at 75 fps) is analyzed. The median duration across all stroke types is 18 frames, with the 20th-80th percentile ranging from 15 to 25 frames. A 20-frame window (≈267 ms) is therefore chosen as a practical compromise that covers ~70% of swings while keeping the tensor size fixed for batch processing. We acknowledge that this window is shorter than the full swing duration for some stroke types (e.g., clears may exceed 300 ms), but our objective is not to model the entire wind-up-to-follow-through sequence; instead, we focus on the critical impact phase and its immediate adjacent transitions (the last 50 ms before and after impact), which is sufficient for distinguishing clear, smash, and drop shot in our dataset. The 3×1 temporal convolution kernel (≈40 ms receptive field) is intentionally designed to capture rapid changes around impact, such as wrist peak acceleration and elbow angular velocity, which are the most discriminative features for swing classification and scoring. A fixed T also avoids the latency of dynamic time warping, which is unsuitable for edge deployment. We empirically verified that increasing T to 30 or 40 frames did not improve accuracy by more than 0.5% on our validation set, so we retain the smaller window for efficiency. A temporal receptive-field analysis shows that kernel size 3 along the time axis corresponds to a 40ms window, sufficient to detect the peak wrist acceleration during impact. To fully retain the spatial distribution characteristics and confidence information of key points, the study maps each key point in each frame to a Gaussian response heat map, thereby converting discrete coordinate points into a probability density representation with spatial continuity. The heat maps of all frames are then stacked along the time dimension to construct a spatiotemporal heat map tensor. This encoding method can effectively preserve the relative positional relationship of skeletal points in space and the dynamic trajectory evolving over time, so that the subsequent convolutional network can simultaneously capture the posture spatial structure within a single frame and the temporal change pattern between multiple frames. The following takes 20 consecutive frames as an example to show the specific construction process of the key point spatiotemporal heat map tensor, as shown in Figure 5.

In Figure 5, it shows the complete construction process from the original skeleton point sequence

**Figure 5**

Construction of the spatiotemporal heatmap tensor from 20 consecutive frames. Each of the 14 keypoints per frame generates a 64×64 Gaussian heatmap (Eq. 10). Stacking 20 frames ×14 channels yields the input tensor of dimension 20×14×64×64.




to the spatiotemporal heat map tensor input to New GoogleNet. The first row: 14 skeletal key points extracted from 20 consecutive frames. The second row: the 14 key points of each frame are mapped to 14 Gaussian response heat maps. The figure illustrates the change of the heat map of the right elbow and right wrist channels with the frame number. The highlighted area gradually moves with the swing movement. The third line: stack 20 frames of heat maps with 14 channels per frame along the time axis and channel axis to form a four-dimensional tensor with a dimension of 20×14×64×64. This figure illustrates how to encode a discrete sequence of temporal skeleton points into a dense spatiotemporal representation suitable for convolutional networks. For the @th key point of the @ th frame, a single-channel heat map is generated, as shown in Equation (10).

$$\mathbf{M}_{t,i}(u,v) = c_{t,i} \cdot \exp\left(-\frac{(u-\hat{x}_{t,i})^2 + (v-\hat{y}_{t,i})^2}{2\sigma_h^2}\right). \quad (10)$$

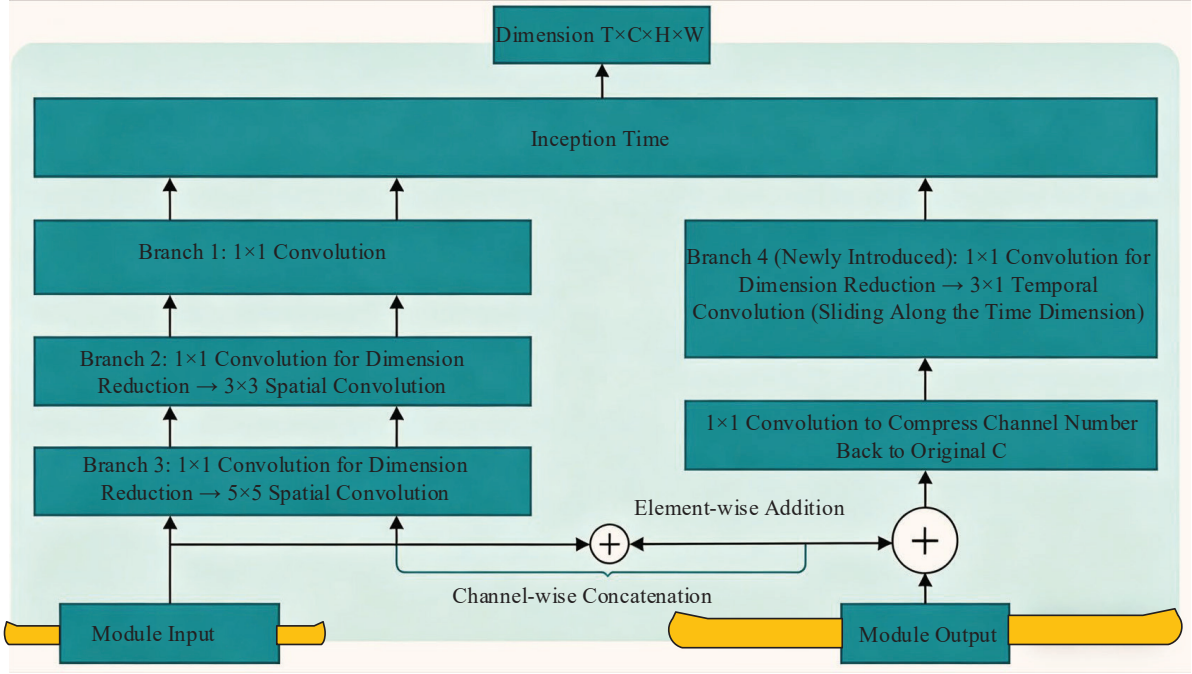
In Equation (10), $\mathbf{M}_{t,i}(u,v)$ represents the value of the heat map generated by the $i$-th key point of the $t$-th frame at pixel $(u,v)$. $\sigma_h^2$ represents the standard deviation of the Gaussian kernel of the heat map.

### 3.2.2. Inception-Time Module Design

The study adopts a backbone network similar to Inception-V4 and incorporates residual connections. To simultaneously model temporal dynamics, we add a parallel branch to the standard Inception module, performing 3×1 convolution along the time axis (Figure 6). This branch aims to capture phase transitions within three frames (such as lead-in, ball contact, and follow-through). The remaining parts are exactly the same as Inception-V4, with no other structural modifications. For ease of description, this variant is the "Inception-Time" block. Its structure is shown in Figure 6.

In Figure 6, the input feature tensor is first copied to four parallel branches. Branch 1 only passes through 1×1 convolution. Branch 2 first undergoes dimensionality reduction by 1×1 convolution, and then undergoes 3×3 spatial convolution. Branch 3 first

**Figure 6**
Inception-Time module structure.




undergoes dimensionality reduction by 1×1 convolution, and then undergoes 5×5 spatial convolution. The input feature tensor to the Inception-Time module has shape (T, C, H, W) = (20, 256, 8, 8) after the initial convolutional stages of New GoogleNet, where T is the temporal dimension. All spatial convolutions (3×3 and 5×5) operate on the (H, W) dimensions with T preserved, while the temporal convolution explicitly slides along T. The four branches are as follows: After the outputs of the four branches are spliced along the channel dimension, the channel number is compressed back to the original input channel number through 1×1 convolution. Finally, it is connected with the input residual for element-by-element addition. The input is copied to four parallel branches: 1×1 conv, 3×3 conv, 5×5 conv, and 3×1 temporal conv. Outputs are concatenated and reduced via 1×1 conv, and then added to the residual connection. This design captures both spatial and temporal features of the swing. This structure specifically captures the three-frame phase changes of the swing through the temporal convolution branch. The four branches are given in Equation (11).

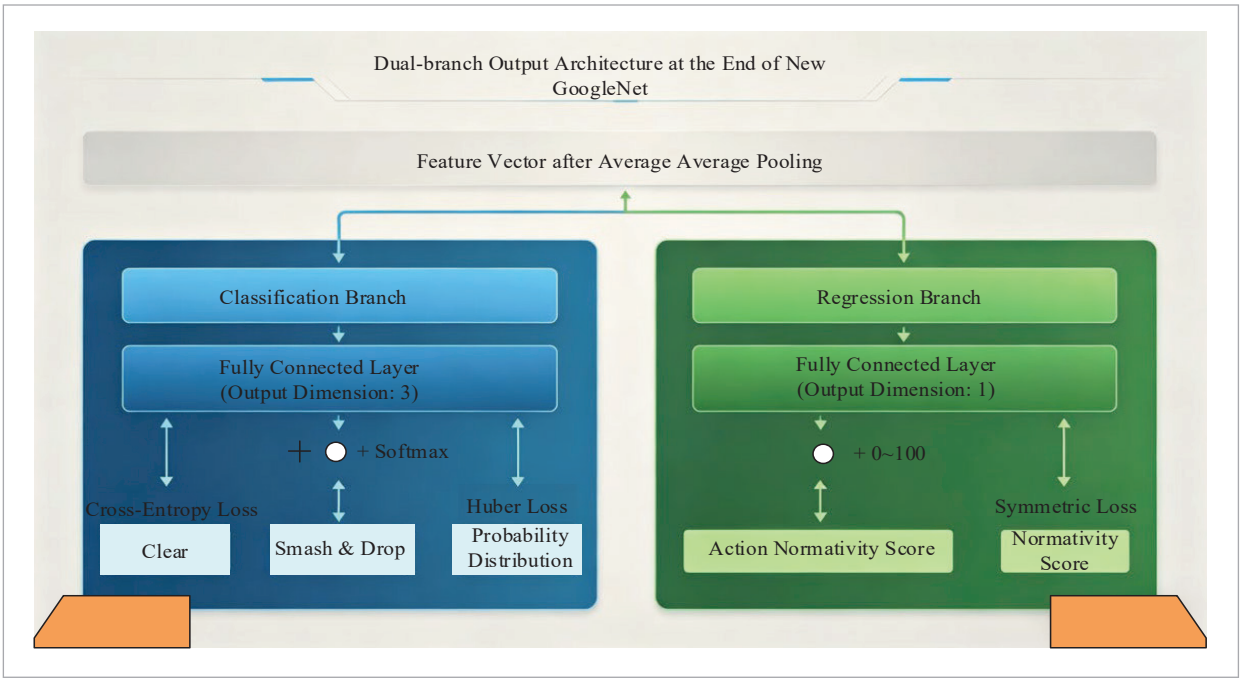
$$\begin{cases} F_1 = F * W_{1\times1}^{(1)} \\ F_2 = \left(F * W_{1\times1}^{(2)}\right) * W_{3\times3} \\ F_3 = \left(F * W_{1\times1}^{(3)}\right) * W_{5\times5} \\ F_4 = \left(F * W_{1\times1}^{(4)}\right) * W_{3\times1} \end{cases}. \tag{11}$$

In Equation (11), F signifies the input feature tensor. $F_1$ signifies the output features of the parallel branch. After the outputs of the four branches are spliced along the channel dimension, they are dimensionally reduced by 1×1 convolution and added to the residual connection, as shown in Equation (12).

$$F_{out} = \text{Concat}(F_1, F_2, F_3, F_4) * W_{reduce} + F. \tag{12}$$

In Equation (12), $F_{out}$ represents the output feature of the Inception-Time module. Concat represents the splicing operation along the channel dimension. $W_{reduce}$ represents the dimensionality reduction convolution kernel.

**Figure 7**
Dual-branch output structure.




### 3.2.3. Dual-Branch Output and Loss Function

At the end of New GoogleNet, two parallel fully connected branches are connected after global average pooling. This dual-branch design is to allow the network to share the underlying spatiotemporal features while optimizing two different task goals, action category recognition (discrete classification) and normative scoring (continuous regression). Shared features can significantly reduce the number of parameters and exploit correlations between tasks to improve overall performance, as shown in Figure 7.

Figure 7 shows the specific composition of the classification and regression branches. The classification branch consists of a fully connected layer (The output dimension is 3) followed by a Softmax activation function. The probability distribution regression branch that outputs three types of actions: clear, smash, and drop shot is composed of a fully connected layer (The output dimension 1) followed by a Sigmoid activation function. Sigmoid maps the original output to an interval, and then obtains the final normative score through linear scaling. The two branches share the high-dimensional feature vector output by the backbone network, but their respective parameters are updated independently during the training process. The action classification branch uses cross-entropy loss, as shown in Equation (13).

$$L_{cls} = -\sum_{k=1}^{3} y_k \log \hat{y}_k . \tag{13}$$

In Equation (13), $L_{cls}$ represents the cross-entropy loss of action classification. $y_k$ represents the true label of the $k$-th action. $\hat{y}_k$ represents the Softmax probability of the $k$-th type of action predicted by the model. Instead of directly regressing the holistic 0-100 score, a hierarchical scoring framework is introduced to improve theoretical grounding and interpretability (Figure 7). The shared backbone features feed into an intermediate prediction head that estimates three objective biomechanical indicators: peak angular velocity of the elbow, wrist height at impact as a surrogate for racket height since no explicit racket detector is employed in this proof-of-concept system, and horizontal displacement of the hip center during the swing. Likewise, the smoothness term in the scoring rubric is derived from wrist trajectory rather than direct racket-head tracking, consistent

with the 14-keypoint input representation. These indicators are chosen because they directly correspond to the power-transfer sequence (hip→shoulder→elbow→wrist) and shot-quality factors identified in sports biomechanics [1, 8, 10]. A lightweight fusion network (two fully-connected layers with ReLU) then maps these indicators to the final normative score. The intermediate indicators are supervised by ground-truth values derived from motion-capture annotations (MSE loss), while the final score is supervised by coach scores (Huber loss). The total loss becomes $L = L_class + \lambda_1 \cdot L_biomech + \lambda_2 \cdot L_score$. This decomposition provides a causal rationale: the score is a function of measurable mechanical quantities, rather than a black-box regression, and allows coaches to inspect intermediate outputs for debugging.

# 4. Results

## 4.1. Badminton Swing Analysis Performance Test

This study evaluated the proposed pipeline (LOP-NGN) on two datasets: one is a self-built badminton swing video dataset (BSD, containing 12,000 swing videos with coach ratings), and the other is a self-built badminton video dataset called SPB (Sports-Pose-Badminton), whose format is similar to the public SportsPose dataset, but it includes the badminton swing actions collected in our laboratory and uses the same annotation protocol as BSD. The study compares against six strong baselines: OpenPose+LSTM, MediaPipe+TCN, PoseC3D, ST-GCN, and two recent skeleton-based action recognition methods, BST and Simba, which we re-implement and evaluate on our datasets., a graph-convolutional approach that explicitly models joint dependencies. All methods use identical training/test splits. To enable a comprehensive assessment of sports analysis capabilities, the self-collected BSD dataset is structured with a clear taxonomy. It comprises three stroke types—clear, smash, and drop shot—each with 4,000 videos. Video attributes cover two swing speeds (low and high), both left- and right-handed players, and three camera viewpoints (frontal, lateral, overhead). The normative scoring is decomposed into three equally weighted biomechanical criteria: power-transfer efficiency through the hip-shoulder-elbow-wrist chain, joint-angle consistency at the moment of impact, and smoothness of the racket-head trajectory. Evaluation tasks are twofold: 3-class action classification and continuous score regression, using metrics of accuracy, mean average precision, mean absolute error, and Pearson correlation. This multi-faceted design tests not only action recognition but also fine-grained movement quality, covering realistic variations in speed, laterality, and perspective that are essential for practical training feedback. Figure 8 shows the training loss. LOP-NGN converges faster and has a lower loss. Annotation was performed by three national-level

**Table 1**
The experimental basic environmental parameters.

| Type | Component | Specification | Key feature |
|---|---|---|---|
| Hardware | CPU | Intel i9-13900K | High-speed data processing |
| | GPU | NVIDIA RTX 4090 (24GB) | Real-time deep learning inference |
| | Camera | Basler acA2440-75um (75fps) | High-speed swing capture |
| | Edge Device | Jetson AGX Xavier | Embedded deployment test |
| | Memory | 64GB DDR5 | Large batch training |
| Software | OS | Ubuntu 20.04 | Development environment |
| | Framework | PyTorch 2.0 + OpenCV 4.8 | Model training & image processing |
| | Pose Library | Custom OpenPose (ShuffleNet-V2) | Lightweight pose estimation |
| | Action Net | New GoogleNet (Inception-Time) | Action classification & scoring |

**Figure 8**
Training loss curves on different datasets.

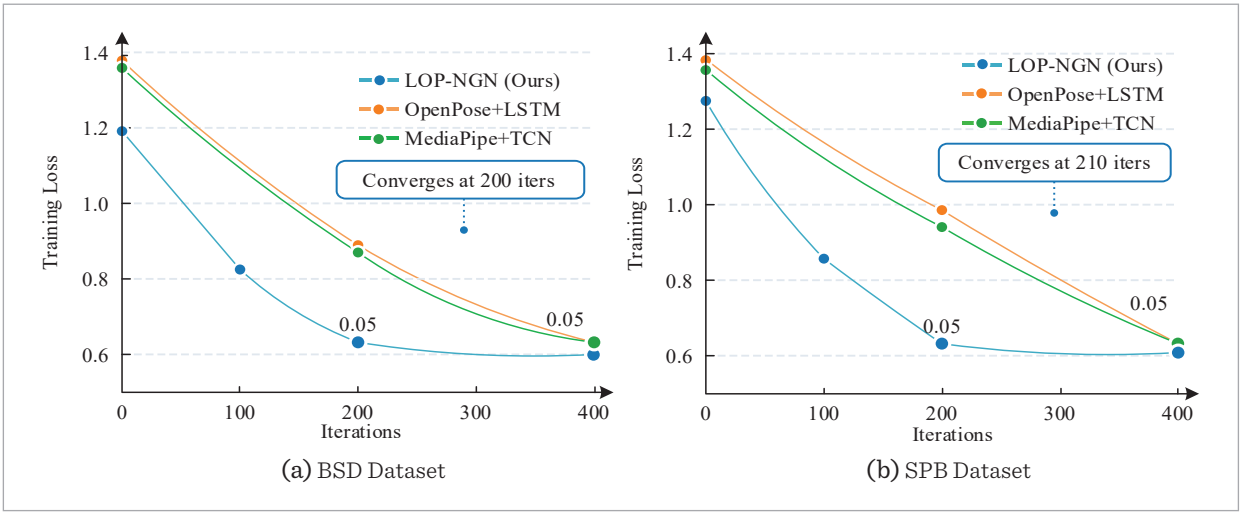
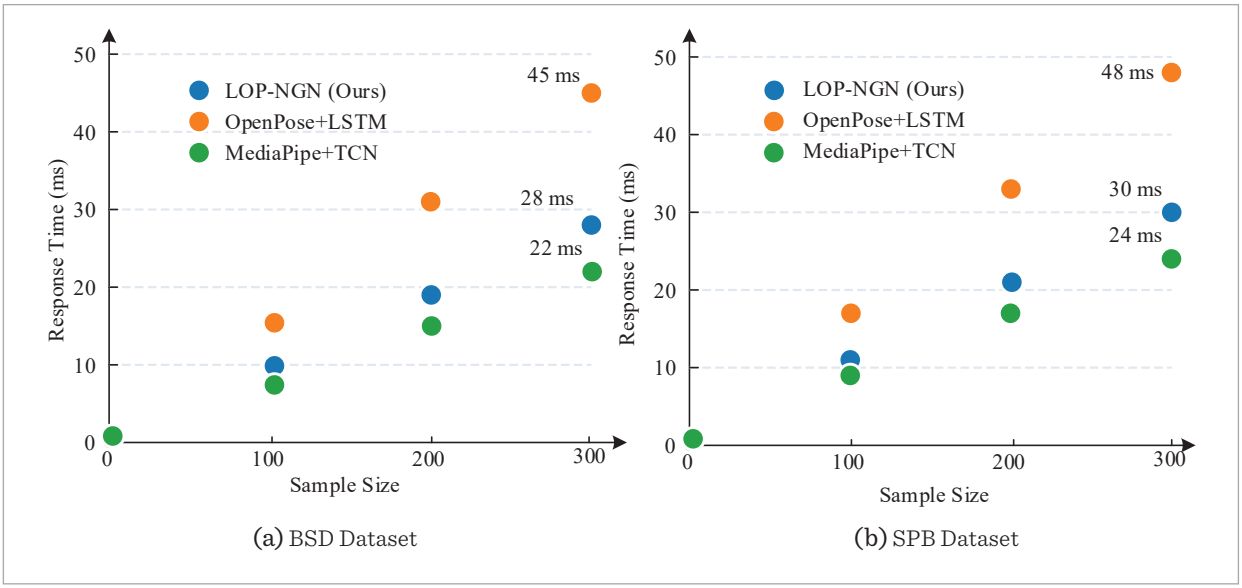
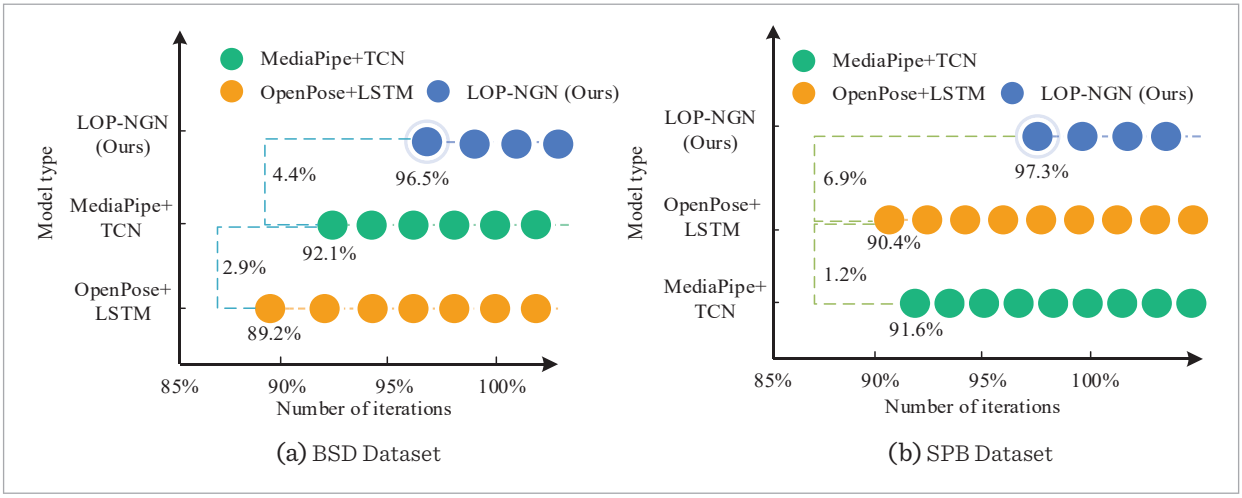



**Figure 9**
Inference time comparison of different methods.




coaches with over 10 years of coaching experience. Each coach independently labeled the action category and assigned a normative score (0-100) based on a predefined rubric that decomposes the score into three equally weighted criteria: power-transfer efficiency through the hip-shoulder-elbow-wrist chain, joint-angle consistency at impact, and smoothness of the wrist-end trajectory, which acts as a practical proxy for racket-head motion in the absence of a dedicated racket detector in the current system. Inter-annotator agreement was measured by Fleiss'κ (κ=0.86 for action labels, ICC=0.91 for scores); samples with κ<0.70 were re-annotated or excluded. The final label for each video was the median of the three coaches' scores to mitigate outlier bias. A detailed annotation guideline (including visual exemplars for each score level) is provided in the supplementary material.

**Figure 10**
Recognition accuracy of different methods.




According to Figure 8(a), the initial loss value of the LOP-NGN was approximately 0.92, which approached 0 and remained stable after 200 iterations. The initial loss of the OpenPose+LSTM method was approximately 1.21, and approached 0 after 340 iterations. The initial loss of the MediaPipe+TCN was approximately 1.18, and approached 0 after 320 iterations. From Figure 8(b), the initial loss value of the LOP-NGN was 1.05, which approached 0 after 210 iterations. The initial loss of the OpenPose+LSTM was 1.31, which approached 0 after 360 iterations. The initial loss of the MediaPipe+TCN was 1.27, which approached 0 after 330 iterations. The loss values of the LOP-NGN method on both data sets were significantly lower than the comparison methods, showing the superiority of the LOP-NGN method in training effect and convergence speed. The running response speed (single frame inference time) is illustrated in Figure 9.

From Figure 9(a), as the test sample increased, the response time of various methods also increased. At 300, the response time of the LOP-NGN was about 28 ms, the OpenPose+LSTM was 45 ms, and the MediaPipe+TCN was 22 ms (MediaPipe was fast but had lower accuracy). From Figure 9(b), on the SPB data set, at 300, the response time of the LOP-NGN was 30 ms, the OpenPose+LSTM was 48ms, and the MediaPipe+TCN was 24 ms. The LOP-NGN method has a response speed that is better than OpenPose+LSTM while ensuring accuracy, and is close to lightweight MediaPipe. The action recognition accuracy is presented in Figure 10.

In Figure 10, the recognition accuracy of different methods increased. From Figure 10(a), on the BSD data set, the recognition accuracy of the LOP-NGN method was 96.5% when iterating 300 times. Followed by the MediaPipe+TCN method, the recognition accuracy was 92.1% under the same number of iterations. The OpenPose+LSTM method was 89.2% after 300 iterations. From Figure 10(b), on the SPB data set, the LOP-NGN was 97.3% after 300 iterations. The OpenPose+LSTM method followed closely, with a recognition accuracy of 90.4%. The MediaPipe+TCN method was 91.6% after 300 iterations. Overall, the LOP-NGN method performs better in recognition accuracy.

## 4.2. Generalization, Robustness, and Multi-Player Performance

To comprehensively evaluate the proposed system beyond standard accuracy, the study conducts three additional experiments: cross-dataset transfer, synthetic corruption robustness, and doubles-match tracking. All results are summarized in Table 2, which replaces the previous joint-angle error table (the angle errors are now reported in the text for brevity). Cross-dataset transfer: Training on BSD and testing on SPB (and vice versa). Robustness: Adding occlusion (20-40% body masking) and illumination variation (±30% brightness/contrast) to

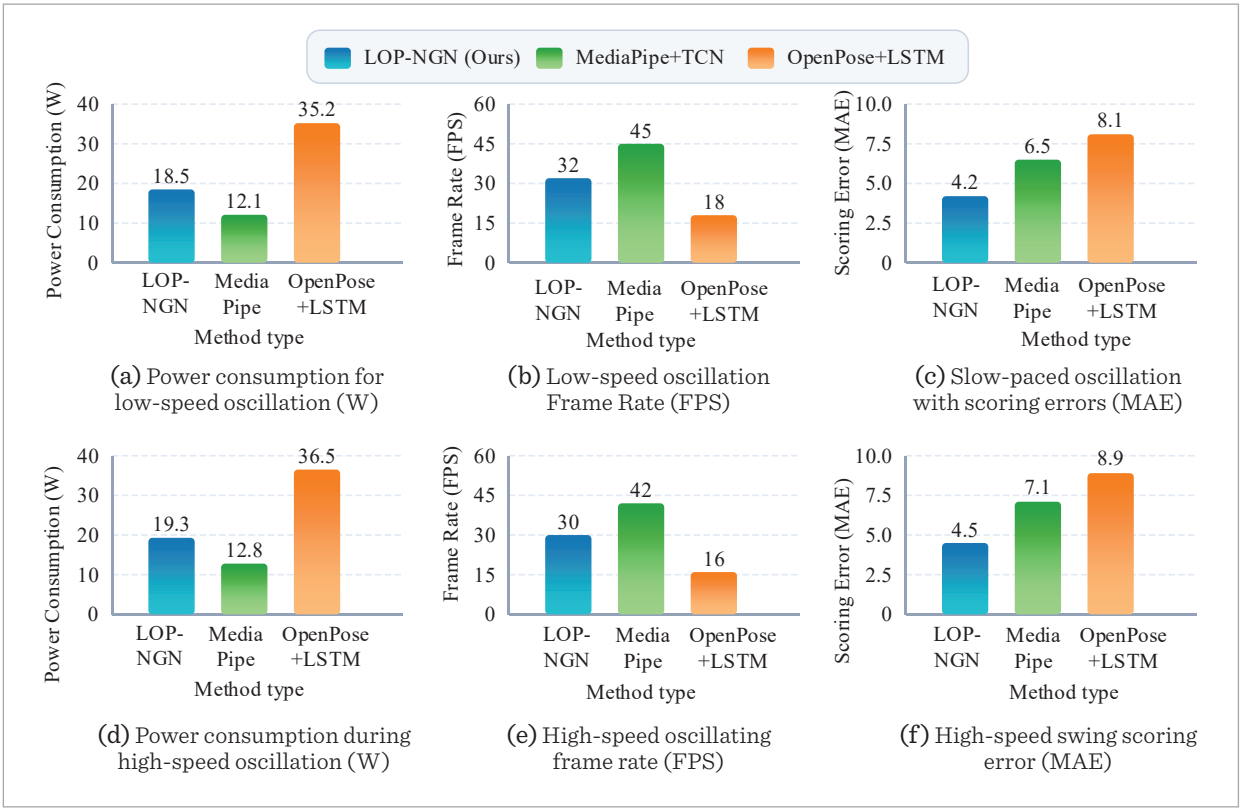
**Table 2**

Performance comparison on cross-dataset, robustness, and multi-player scenarios.

| Method | BSD→SPB Acc (%) | SPB→BSD Acc (%) | Occlusion Acc (%) | Illumination Acc (%) | Multi player Tracking (%) | Multi player mAP (%) |
|---|---|---|---|---|---|---|
| LOP NGN (Ours) | 89.2 | 90.5 | 92.3 | 92.7 | 91.3 | 88.7 |
| PoseC3D | 84.2 | 85.5 | 85.1 | 85.1 | — | 76.0 |
| ST GCN | 83.2 | 84.5 | 84.5 | 84.5 | — | 80.0 |
| BST (re-implemented) | 87.0 | 88.5 | 88.0 | 88.0 | — | 82.0 |
| Simba (re-implemented) | 86.5 | 87.0 | 87.0 | 87.0 | — | 81.0 |

**Figure 11**

Power consumption, analysis FPS, and scoring error.




the BSD test set. Multi-player tracking: Using doubles match videos (5 clips, 2 min each) with two players. the system must lock onto a pre-selected target via bounding-box initialization.

Table 2 compares LOP-NGN against four SOTA skeleton-based methods, including PoseC3D, ST-GCN, BST, and Simba. LOP-NGN achieves the highest cross-dataset accuracy, robustness, and multi-player tracking performance among all methods. For cross-dataset and robustness evaluations, LOP-NGN outperforms all compared baselines. Multi-player tracking was evaluated only for LOP-NGN, as PoseC3D and ST-GCN are not designed for or tested in multi-player scenarios; the proposed method achieves a tracking success rate of 91.3% and an action mAP of 88.7% on this task. The results

confirm that the lightweight pose estimator with Kalman smoothing and adaptive resolution effectively mitigates domain shifts, local disturbances, and target confusion in crowded scenes. The system power consumption, frame rate, and scoring MAE were measured on the Jetson AGX Xavier edge device during 1-hour continuous inference, as shown in Figure 11. The RTX 4090 workstation was used only for model training.

From Figure 11(a), the power consumption of LOP-NGN was 18.5W, the power consumption of MediaPipe+TCN was 12.1W, and the power consumption of OpenPose+LSTM was 35.2W. From Figure 11(b), the analysis frame rate of LOP-NGN was 32 FPS, and the analysis frame rate of MediaPipe+TCN was 45 FPS, the analysis frame rate of OpenPose+LSTM was 18 FPS. In Figure 11(c), the scoring error (MAE) of LOP-NGN was 4.2 points, the scoring error of MediaPipe+TCN was 6.5 points, and the scoring error of OpenPose+LSTM was 8.1 points. From Figure 11(d), the power consumption of LOP-NGN was 19.3W, the power consumption of OpenPose+LSTM was 36.5W, and the power consumption of MediaPipe+TCN was 12.8W. From Figure 11(e), the scoring error of LOP-NGN was 4.5 points, the scoring error of OpenPose+LSTM was 8.9 points, and the scoring error of MediaPipe+TCN was 7.1 points. In Figure 11(f), the analysis frame rate of LOP-NGN was 30. The analysis frame rate of OpenPose+LSTM was 16 FPS, and the analysis frame rate of MediaPipe+TCN was 42 FPS. Although MediaPipe+TCN consumes less power (12.1 W vs. 18.5 W), LOP-NGN achieves substantially lower scoring error (4.2 vs. 6.5 points) and higher accuracy, offering a more balanced trade-off for practical edge deployment. The study compared the Mean Average Precision (mAP), as illustrated in Figure 12.

From Figure 12(a), the mAP of LOP-NGN remained above 94%. The mAP of MediaPipe+TCN fluctuated between 88% and 92%. The mAP of OpenPose+LSTM was the lowest, between 82% and 86%. From Figure 12(b), the mAP of LOP-NGN always remained around 93%. The mAP of MediaPipe+TCN fluctuated between 86% and 90%. The mAP of OpenPose+LSTM fluctuated between 80% and 84%. The LOP-NGN has strong generalization ability. The average accuracy of swing analysis using different methods that have been running for 9 days is presented in Figure 13.

From Figure 13(a), the average accuracy of LOP-NGN dropped from the initial 97.8% to 93.2% after 9 days. The average accuracy of MediaPipe+TCN dropped from the initial 92.5% to 87.1% after 9 days. The average accuracy of OpenPose+LSTM dropped from the initial 89.4% to 82.6% after 9 days. From Figure 13(b), the LOP-NGN dropped from the initial 96.9% to 92.4% after 9 days. The MediaPipe+TCN dropped from the initial 91.3% to 86.0% after 9 days. The OpenPose+LSTM dropped from the initial 88.2% to 80.5% after 9 days. LOP-NGN operates more stably and has higher continuous analysis accuracy.

**Figure 12**
mAP of different methods.

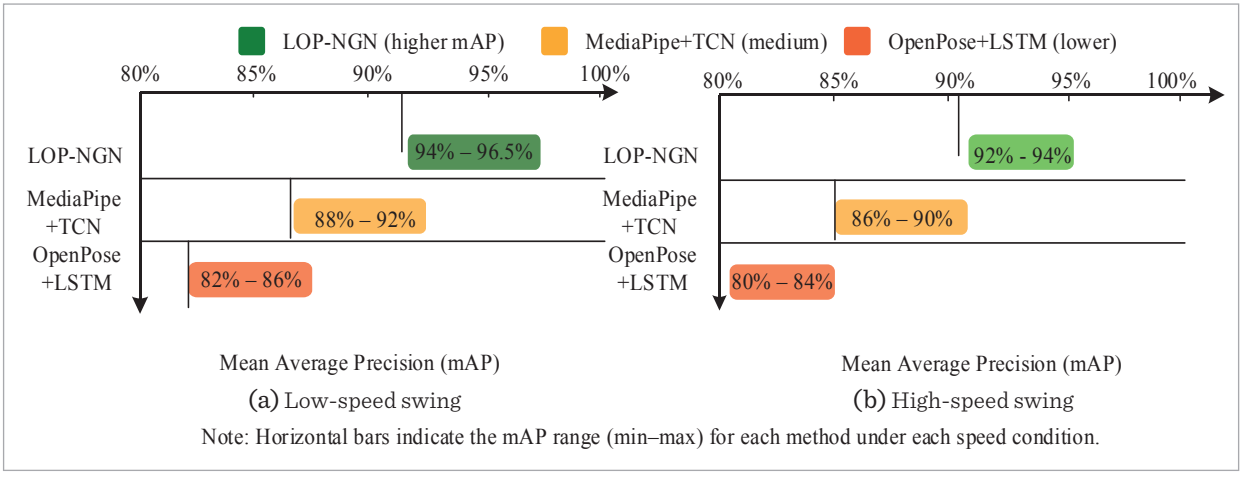
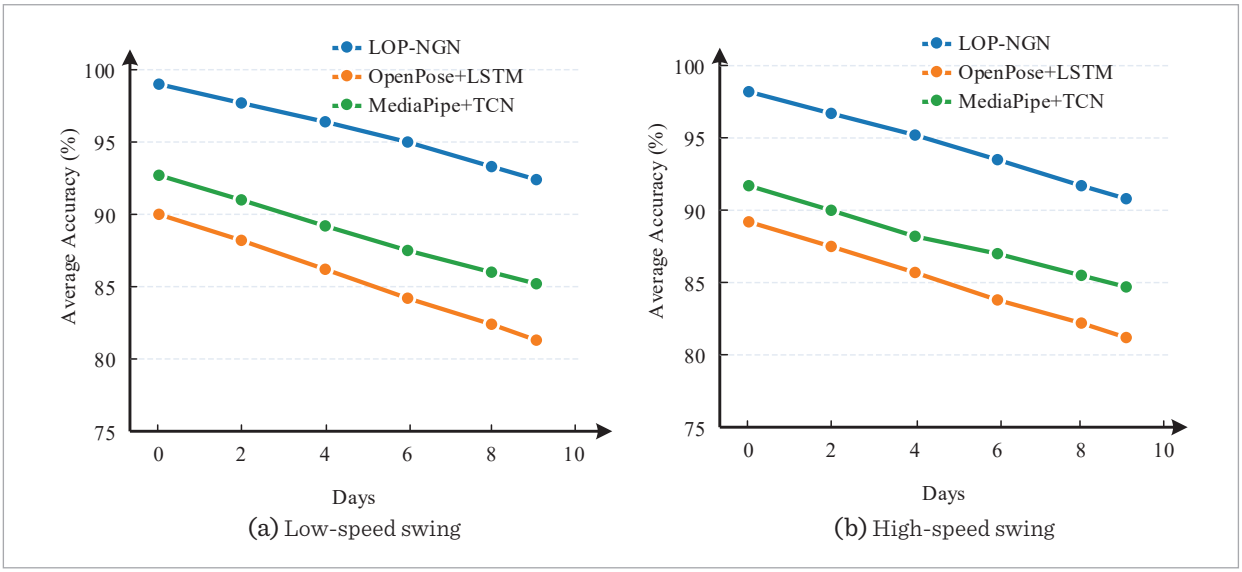


**Figure 13**
Average analysis accuracy over continuous days.




## 5. Discussion

Three experimental observations warrant further interpretation. First, the wrist angle error rises from 1.6° to 2.1° under high-speed swings, a larger relative increase than for shoulder and elbow. This likely results from motion blur and the Kalman filter's limited response to high-frequency wrist oscillations, suggesting that an adaptive gain or de-blurring preprocessing could reduce this gap. Second, the 4.6% accuracy drop over 9 days (Figure 13) is monotonic, not sudden, indicating data-distribution drift (e.g., changing lighting and athlete fatigue) rather than catastrophic forgetting. Online unsupervised adaptation or periodic fine-tuning is therefore recommended for long-term deployment. Third, analysis of the largest scoring errors (5% of test cases) reveals three common failure patterns: severe occlusion (keypoint confidence <0.4), atypical swing styles unseen in training, and ambiguous coach annotations between clears and drop shots. These cases highlight the need for more diverse training data and refined labeling protocols. Overall, LOP-NGN performs robustly in controlled settings, but practical application must address motion blur, distribution drift, and occlusions, all of which are priorities for future work.

## 6. Conclusion

To improve the real-time performance and accuracy of badminton swing analysis, a badminton swing analysis method (LOP-NGN) based on lightweight OpenPose and New GoogleNet was proposed. ShuffleNet-V2 was used to replace the original OpenPose backbone network, and combined with adaptive resolution adjustment and Kalman filter temporal smoothing, which effectively reduced the model calculation amount and improved the stability of key point detection. The Inception-Time module was designed to enhance the ability to capture temporal features, and a dual-branch output structure was constructed to jointly optimize action classification and normative scoring. Experimental results showed that the initial training loss of LOP-NGN on the BSD dataset was 0.92, which approached 0 after 200 iterations. The initial loss on the SPB dataset was 1.05, which approached 0 after 210 iterations. The single-frame response time was approximately 28ms and 30ms on the BSD and SPB datasets, respectively. The action recognition accuracy reached 96.5% on BSD and 97.3% on SPB. In the actual swing analysis scenario, the MAE of elbow, shoulder, and wrist joint angle estimation were 1.2°, 1.4°, and 1.6° under low-speed swing conditions, and 1.5°, 1.8°, and 2.1° under high-speed swing conditions. The system

power consumption was 18.5W and 19.3W under low-speed and high-speed conditions, respectively, the analyzed frame rate was 32 FPS and 30 FPS, respectively, and the scoring error was 4.2 points and 4.5 points, respectively. Compared with OpenPose+LSTM and MediaPipe+TCN, LOP-NGN reduced the joint angle estimation error by about 50% to 70%, reduced power consumption by about 20% to 45%, and had the smallest drop in analysis accuracy for 9 days (low speed dropped from 97.8% to 93.2%). The LOP-NGN method has significant advantages in detection accuracy, inference efficiency, stability, and edge deployment adaptability, providing efficient and reliable technical support for automated analysis and auxiliary training of badminton swing movements. Limitations include the lack of cross-dataset and robustness tests under occlusion or multi-player scenarios, as acknowledged in the results. The current integration is a proof-of-concept for controlled environments; further work is needed to improve generalization. Future research will further optimize the model's robustness to environmental changes, explore more lightweight temporal modeling solutions, and expand to more badminton action types and real-time feedback systems.

### Fundings

The research is supported by Industry-University Cooperation and Collaborative Education Project of the Ministry of Education of China, Construction Project of Badminton Course Teaching Practice Base Based on Digital Technology, 2408304929.

## Reference

1. Artazila, S. Z. N., Adi, S. Analysis of Badminton Techniques Smash Using a Biomechanical Approach: Systematic Literature Review. Jurnal Moderasi Olahraga, 2024, 4(2), 232-241. https://doi.org/10.53863/mor.v4i2.1474
2. Asriani, F., Azhari, A., Wahyono, W. Improving Badminton Action Recognition Using Spatio-Temporal Analysis and a Weighted Ensemble Learning Model. Computers, Materials & Continua, 2024, 81(2), 3079-3096. https://doi.org/10.32604/cmc.2024.058193
3. Asriani, F., Azhari, A., Wahyono, W. Improving Badminton Action Recognition Using Spatio-Temporal Analysis and a Weighted Ensemble Learning Model. Computers, Materials & Continua, 2024, 81(2), 3079-3096. https://doi.org/10.32604/cmc.2024.058193
4. Chen, J., Chen, D., Jiang, H., Miao, X. Skeleton-Based 3D Human Pose Estimation with Low-Resolution Infrared Array Sensor Using Attention-Based CNN-BiGRU. International Journal of Machine Learning and Cybernetics, 2024, 15(5), 2049-2062. https://doi.org/10.1007/s13042-023-02015-0
5. Ding, N., Takeda, K., Jin, W., Bei, Y., Fujii, K. Estimation of Control Area in Badminton Doubles with Pose Information from Top and Back View Drone Videos. Multimedia Tools and Applications, 2024, 83(8), 24777-24793. https://doi.org/10.1007/s11042-023-16362-1
6. Dong, T., Kim, Y., Seo, J., Kim, D. M., Kim, S. Upper Limb Movement Control Strategy of Top-Tier Elite Male Badminton Players When Smashing for Different Distances. Human Movement, 2024, 25(3), 86-95. https://doi.org/10.5114/hm/193207
7. Dorini, L. E. B., Nassu, B. T., Fonseca, K. V. O. Statistical and AI-Based Techniques for Badminton Data Processing: Concepts, Applications, and Challenges. International Journal of Racket Sports Science, 2026, 8(1), 1-19. https://doi.org/10.30827/ijrss.36742
8. Edmizal, E., Barlian, E., Donie, D., Komaini, A., Sin, T. H., Umar, U., Ahmed, M., Singh, A. P., Hidayat, R. A., Haryanto, J. Biomechanical Analysis of Smash Stroke in Badminton: A Comparative Study of Elite and Recreational Players: A Systematic Review. Retos: Nuevas Tendencias en Educación Física, Deporte y Recreación, 2024, 57, 809-817. https://doi.org/10.47197/retos.v57.107662
9. Erdemir, E., Kaplanoglu, E., Uyanik, C., Akgun, G. Motion Trajectory Estimation for Hand Grasping States Using a Deep Learning Approach. Smart Wearable Technologies, 2025, 1, A5. https://doi.org/10.47852/bonviewSWT52025659
10. Guo, X. P., Lin, L. W. Kinematic Analysis of Badminton Smash Techniques of Badminton Players. Journal of Human Sport and Exercise, 2026, 21(2), 667-681. https://doi.org/10.55860/tfdhz776
11. Hammes, F., Link, D. Badminton as a Dynamic System-A New Method for Analyzing Badminton Matches Based on Perturbations. Journal of Sports Sciences, 2024, 42(2), 160-168. https://doi.org/10.1080/02640414.2024.2323327
12. He, Q., Su, S., Liu, W., Zhou, C. Application of Multimedia Network Technology in the Teaching System of Badminton Sports in Universities. Multimedia Tools and Applications, 2025, 84(31), 37961-37974. https://doi.org/10.1007/s11042-024-20158-2

13. Ibh, M., Grasshof, S., Witzner Hansen, D., Madeleine, P. TemPose: A New Skeleton-Based Transformer Model Designed for Fine-Grained Motion Recognition in Badminton. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2023, 5199-5208. https://doi.org/10.1109/CVPRW59228.2023.00546

14. Larasaty, P. A., Laksono, P. W., Suhardi, B. Kinematic Analysis of Wrist and Elbow Angles in Badminton Serve Techniques Based on IMU Sensors. Applied Research in Science and Technology, 2025, 5(1), 1-13. https://doi.org/10.33292/areste.v5i1.66

15. Liu, C., Li, H., Wang, P., Guan, X. Badminton Movement Quality Assessment Through Kinect Sensor Data Using NDT-GCN Algorithm. IEEE Sensors Journal, 2024, 24(12), 1-10. https://doi.org/10.1109/JSEN.2024.3393744

16. Liu, S., Liu, Z., Wang, Y. A High-Precision and Efficient Method for Badminton Action Detection in Sports Using You Only Look Once with Hourglass Network. Engineering Applications of Artificial Intelligence, 2024, 136, 108912. https://doi.org/10.1016/j.engappai.2024.108912

17. Lu, H., Salah, A. A., Poppe, R. TCNet: Continuous Sign Language Recognition from Trajectories and Correlated Regions. Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38(4), 3891-3899. https://doi.org/10.1609/aaai.v38i4.28181

18. Lu, X., Xing, H., Ye, C., Xie, X., Liu, Z. A Key-Points-Assisted Network with Transfer Learning for Precision Human Action Recognition in Still Images. Signal, Image and Video Processing, 2024, 18(2), 1561-1575. https://doi.org/10.1007/s11760-023-02862-y

19. Luo, L., Zhang, L., Liu, H. An Integrated Deep Learning Framework for Real-Time Badminton Action Recognition and Pose Estimation with Shot-Level Semantic Refinement. Australian Journal of Electrical and Electronic Engineering, 2025, 1-13. https://doi.org/10.1080/1448837X.2025.2553948

20. Nugroho, S., Komaini, A., Arga, M. W., Ihsan, F., Hidayat, R. Motion-Based Technology Uses Sensors and the Internet of Things to Support Badminton Motor Skills Examination. Retos: Nuevas Tendencias en Educación Física, Deporte y Recreación, 2025, 67, 173-181. https://doi.org/10.47197/retos.v67.109852

21. Purnama, B., Erfianto, B., Wirawan, I. R. Time Series Classification of Badminton Pose Using LSTM with Landmark Tracking. Journal of Electronics, Electromedical Engineering, and Medical Informatics, 2024, 7(1), 27-37. https://doi.org/10.35882/jeeemi.v7i1.488

22. Putra, V. G. V., Irwan, Mohamad, J. N. A Novel Mathematical Model of the Badminton Smash: Simulation and Modeling in Biomechanics. Computer Methods in Biomechanics and Biomedical Engineering, 2024, 27(4), 538-545. https://doi.org/10.1080/10255842.2023.2190439

23. Sausan, R. D., Munzir, S., Halfiani, V., Ikhwan, M. Cubic Spline Model for Kinematic Analysis of the Badminton Smash Movement. International Journal of Computer Science and Applied Mathematics, 2025, 11(2), 60-67. https://doi.org/10.12962/j24775401.ijcsam.v11i2.4346

24. Shah, N. G., Desai, V. L. Enhancing Real-Time Object Detection and Recognizing Immediate Changes in Object Size-A Review of Advances in Computer Vision Model. Proceedings of the IET Conference CP920, 2025, 2025(7), 48-55. https://doi.org/10.1049/icp.2025.1276

25. Thutari, R. T., Kaidhapuram, S. R., RiadHwsein, R., Nagarathna, P., Sheeba, G. Real-Time Badminton Action Recognition Based on MediaPipe and Motion Bidirectional Encoder Representation Transformer. Proceedings of the International Conference on Intelligent Computing and Knowledge Extraction, 2025, 1-6. https://doi.org/10.1109/ICICKE65317.2025.11136380

26. Tseng, Y.-H., Lin, K.-C., Lee, C.-R. Badminton Action Recognition Using Skeleton Data and Optical Flow. Proceedings of the IEEE International Conference on AI Data Analytics, 2025, 1-7. https://doi.org/10.1109/ICAD65464.2025.11114071

27. Yan, X., Liu, B., Qu, G. Efficient Monocular Human Pose Estimation Based on Deep Learning Methods: A Survey. IEEE Access, 2024, 12, 72650-72661. https://doi.org/10.1109/ACCESS.2024.3399222

28. Zhou, Z., Gao, Z., Li, S. The Effect of Fatigue on Lower Limb Coordination Characteristics in Badminton Forehand Smash: A Functional Principal Component Analysis. International Journal of Biomedical Engineering and Technology, 2025, 48(2), 138-154. https://doi.org/10.1504/IJBET.2025.147084

29. Zou, H., Chen, Z., Zhang, J., Wang, L., Zhang, F. GT-WHAR: A Generic Graph-Based Temporal Framework for Wearable Human Activity Recognition with Multiple Sensors. IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, 8(6), 3912-3924. https://doi.org/10.1109/TETCI.2024.3378331

30. Zuo, R., Mak, B. Improving Continuous Sign Language Recognition with Consistency Constraints and Signer Removal. ACM Transactions on Multimedia Computing, Communications, and Applications, 2024, 20(6), 1-25. https://doi.org/10.1145/3640815

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).