Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA37518

A Badminton Hitting Recognition Method Based on Posture Keypoints

Yichen Zhuang

College of Publishing, University of Shanghai for Science and Technology, Shanghai, China
zyccyz2891@gmail.com

Abstract. Automatically identifying which stroke a badminton player has performed is highly useful for recording their technical statistics and supporting tactical analysis. Most current analysis tools only recognize strokes at a coarse level and cannot tell apart closely related stroke types, nor can they reliably label the exact category of a given hit on their own. To tackle this problem, this paper designs a lightweight Transformer stroke classification method based on player pose keypoints. The method first uses You Only Look Once Pose (YOLO-Pose) to extract 17 skeletal keypoints of players from match videos, constructs 37-dimensional frame-level features through court coordinate mapping and movement speed calculation, and then uses a three-layer Transformer Encoder to classify the temporal feature sequences and output four stroke types: smash, clear, drop and net shot. Evaluated on ShuttleSet, the model attains 86.9% test accuracy and a 96.4% F1 score on net shots. The results show that coarse-grained stroke recognition can be effectively accomplished using only skeleton information. In later stages shuttlecock and racket cues would be added so that even finer stroke differences can be told apart.

Keywords: Stroke classification, Pose estimation, Transformer, Computer vision

1. Introduction

A badminton rally moves very quickly, and players often flip between attack and defense in a matter of seconds. Every stroke a player chooses carries a tactical purpose. Over the past years, computer vision has become a common tool for analyzing sports scenes [1]. If these stroke actions can be detected automatically, coaches and players gain an objective record of technique, which lets them plan training and adjust match strategy on solid evidence.

Today's mainstream badminton analysis platforms are already able to detect player poses, map the court, and track trajectories; open-source projects such as Good-Badminton can even turn match videos into visual statistics like heatmaps and scatter plots [2]. In academic research, there have been several datasets and methods for badminton tactical analysis, for example the ShuttleSet dataset, which provides stroke-level annotations (stroke type, stroke timing and court coordinates) for professional matches, laying the data foundation for quantitative stroke analysis [3]. On this basis, ShuttleNet further evolved into a position-aware encoder-decoder model that predicts the stroke type of each hit by fusing rally progress and player style information. Li et al. constructed the VideoBadminton video dataset and trained video recognition models to complete badminton action

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BA37518

recognition. Even so, most of this work stays at the stage of perception, prediction, or broad action recognition; the task of splitting stroke actions into fine-grained types purely from skeleton keypoints has not been studied much. Accordingly, this study builds a lightweight Transformer classifier that works from pose keypoints. The 17 keypoints produced by You Only Look Once Pose (YOLO-Pose) are first projected onto the court coordinate system, then merged with movement speed into temporal features; a three-layer Transformer Encoder finally maps these features to one of the four stroke types.

2. Related work

2.1. Human pose estimation

The goal of human pose estimation is to locate keypoints inside an image. Deep-learning approaches generally fall into two families: top-down and bottom-up [4]. YOLO-Pose incorporates pose estimation into the unified YOLO framework, and performs excellently with efficient inference on the Common Objects in Context (COCO) dataset [5]. Real-Time Multi-Person Pose Estimation (RTMPose) balances accuracy and speed based on heatmap regression [6].

This paper selects YOLO-Pose as the front-end feature extractor. It directly outputs the 17 COCO-standard keypoints, has fast inference speed, and is suitable for the batch processing requirements of entire match videos. In sports action analysis scenarios, 17 keypoints are already sufficient to cover the main body parts.

2.2. Skeleton-based action recognition

For recognizing actions, skeleton keypoint sequences are a compact representation that holds up well against changes in lighting and background. In earlier studies, Long Short-Term Memory (LSTM) networks were the common choice for modeling such temporal sequences [7]. Spatial Temporal Graph Convolutional Network (ST-GCN) introduced graph convolution to model the relations of skeleton nodes in both spatial and temporal dimensions [8]. In recent years, Transformer has been widely used in action recognition thanks to its long-range dependency modeling capability [9, 10].

Transformer is not limited by the sequential recursive structure, can directly capture global temporal dependencies, and has high training parallelism. This paper selects a lightweight Transformer as the classification backbone, taking into account both long-range temporal modeling capability and parameter control.

2.3. Badminton video analysis

Good-Badminton is a complete open-source system in the field of badminton video analysis, covering pose detection, court annotation, player tracking and trajectory visualization. Its pipeline provides the basis for pose extraction and coordinate mapping for this paper.

Released at KDD 2023, the dataset called shuttleset collects match videos of five professional games and attaches stroke-level labels to them—covering stroke type, timing, and court position for every hit. On this basis, ShuttleNet uses rally progress and player style information to predict stroke types, while VideoBadminton provides a video dataset for badminton action recognition [11, 12]. The training and evaluation in this paper are both carried out based on this dataset.

DOI: 10.54254/2755-2721/2026.BA37518

3. Method

3.1. System framework

After the match video is input, YOLO-Pose extracts the skeletal keypoints of the two players frame by frame. The court annotation module establishes a perspective mapping from image coordinates to standard court coordinates to achieve feature normalization.

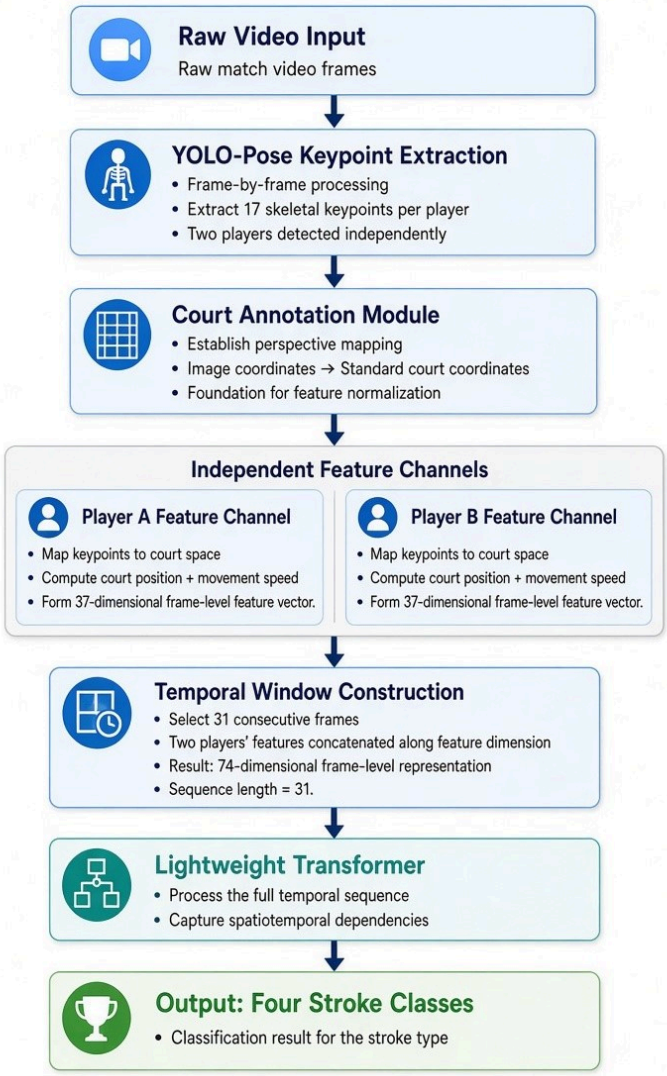


Figure 1. Overall system pipeline (picture credit: original)

After the keypoints of each player are mapped to the court space, the position and movement speed are combined to form 37-dimensional frame-level features. Using a 31-frame temporal window, the features of the two players are concatenated along the dimension axis to form a 74-dimensional frame-level representation. After the sequence is fed into the Transformer, four types of stroke results are output. The overall system pipeline is shown in Figure 1.

DOI: 10.54254/2755-2721/2026.BA37518

3.2. Pose detection

YOLO-Pose is based on the Ultralytics framework, and produces detection boxes and 17 keypoints (x,y) with confidences simultaneously through multi-task learning. The keypoints cover the nose, eyes, ears, shoulders, elbows, wrists, hips, knees and ankles, following the COCO standard definition.

Detection needs to filter out non-player targets. Only individuals located within the court region are retained. The player position is determined by the midpoint of the two ankle keypoints (indices 15 and 16), and a 10-pixel offset is applied. This strategy is more reasonable than using the torso center, because the foot position directly reflects the stance and the center of gravity distribution.

Selecting YOLO-Pose is mainly based on the trade-off between inference efficiency and output format. It can achieve near-real-time inference while maintaining high accuracy, which is crucial for processing entire match videos. Its COCO-standard output requires no additional keypoint conversion.

3.3. Feature construction

The raw keypoint coordinates are in the image pixel space and are greatly affected by the camera viewpoint. This paper maps the coordinates to the court coordinate system (unit: meter) through a perspective transformation matrix, with the origin set at the bottom-left corner of the court.

$$(x_c, y_c) = H \cdot (x_i, y_i) \tag{1}$$

Each frame's feature vector f_t is composed of three types of information. The court coordinates (x,y) of the 17 keypoints, 34 dimensions in total, form the skeleton. The court coordinates of the midpoint of the player's two feet, 2 dimensions, provide spatial position information. The movement speed, 1 dimension, is the Euclidean distance between consecutive frames divided by the time interval, capturing the instantaneous motion intensity.

The features of the two players are concatenated along the dimension axis in each frame to form F_t. A temporal window of 31 frames corresponds to about 1.24 seconds (25fps), taking 15 frames before and after the stroke frame, covering the preparation and follow-through phases of the stroke.

Mapping coordinates onto the court weakens the influence of the camera angle, while speed reflects how much power a stroke carries. Put together, these three cues already give the classifier the key evidence it needs to tell strokes apart.

A velocity term contained by this paper because stroke actions are inherently dynamic: a smash moves much faster than a drop, and that difference shows up clearly in how much the skeleton shifts from one frame to the next.

3.4. Transformer classification model

The 74-dimensional input features are projected to a 128-dimensional space by a linear layer, and after adding sinusoidal positional encoding, they are processed by a three-layer Transformer Encoder. The encoder output is mapped to the four classes through mean pooling and a fully connected layer. The detailed architecture is shown in Figure 2.

DOI: 10.54254/2755-2721/2026.BA37518

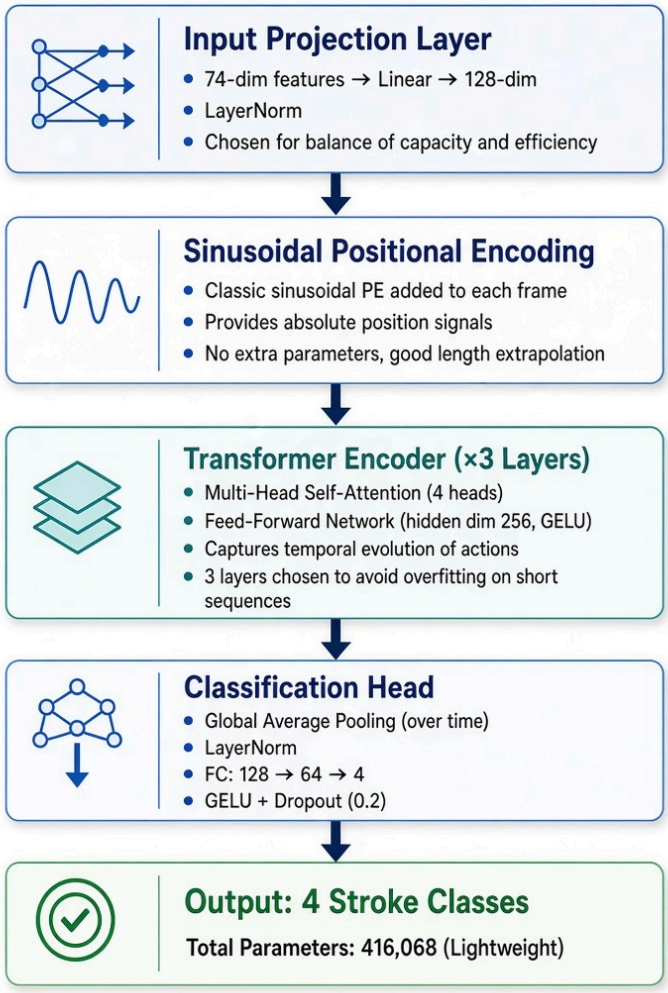


Figure 2. Architecture of the Transformer classification model (picture credit: original)

The network is built from three parts. First, an input projection layer lifts the 74-dimensional input into a 128-dimensional space and applies Layer Normalization. This size represents a deliberate trade-off, maintaining sufficient expressive power while concurrently reducing the computational cost.

Sinusoidal positional encoding is used to provide frame-order information for the sequence, which introduces no extra parameters and has a certain extrapolation ability for unseen sequence lengths [9].

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{\frac{2i}{d}}}), PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{\frac{2i}{d}}}) \tag{2}$$

The encoding result is added to the projected features and then fed into the Transformer Encoder. Each layer contains 4-head self-attention and a feed-forward network (hidden dimension 256, Gaussian Error Linear Unit (GELU) activation). Self-attention enables the model to freely attend to information at different frames within the window.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

DOI: 10.54254/2755-2721/2026.BA37518

Stacking 3 encoder layers is a balance based on parameters and performance. The temporal hierarchy of the 31-frame window is not complex, and overly deep networks are prone to overfitting on small samples.

The encoder outputs are mean-pooled along the temporal dimension to compress the information of the whole window into a single vector. After LayerNorm, they are mapped to the four classes through a 128-64-4 fully connected layer. GELU activation and Dropout(0.2) are added to the middle layers. In total, the model contains 416,068 parameters.

4. Experiments and results

4.1. Dataset

ShuttleSet contains stroke-level annotations of five professional matches [3]. After processing with YOLO-Pose, pose detection results for 200,309 frames are generated. The original 18 stroke classes are merged into four classes.

After preprocessing and alignment, a total of 2,655 valid samples are obtained. The distribution of each stroke category is shown in Figure 3.

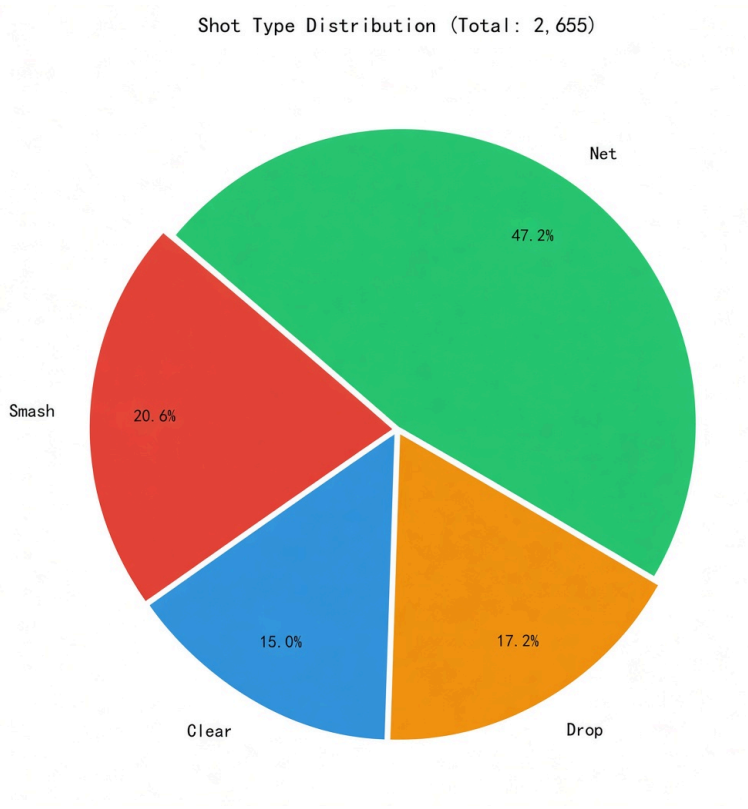


Figure 3. Category distribution of the dataset (picture credit: original)

The samples are significantly imbalanced, with net shots far exceeding the other classes, reflecting the natural distribution of professional matches. Training uses weighted cross-entropy loss, with weights inversely proportional to class frequency, strengthening the loss contribution of minority-class samples.

DOI: 10.54254/2755-2721/2026.BA37518

4.2. Experimental setup

The AdamW optimizer is selected, which decouples weight decay from adaptive learning rates and generalizes better on Transformer-type models. The learning rate adopts cosine annealing (T_0=10) with warm restarts. The batch size is 64, and the model is trained for 100 epochs.

Training is performed on an NVIDIA RTX 3070 Ti (8GB). Data augmentation includes random Gaussian noise (probability 0.3), random scaling (0.9-1.1x) and temporal offset jitter (±3 frames). The checkpoint with the highest validation accuracy is taken as the final model. The training hyperparameters are set as shown in Table 1 below.

Table 1. Training hyperparameter settings

Hyperparameter	Value
Optimizer	AdamW
Initial learning rate	10^{-3}
Weight decay	10^{-4}
Batch size	32
Number of epochs	80
Learning rate schedule	CosineAnnealingWarmRestarts (T_0=10)
Loss function	Weighted cross-entropy

4.3. Overall performance

The model converges quickly in the first 20 epochs; the best validation accuracy of 89.46% appears at epoch 78, and the test accuracy is 86.91%. The gap between the two is about 2.6 percentage points, which is within a reasonable range.

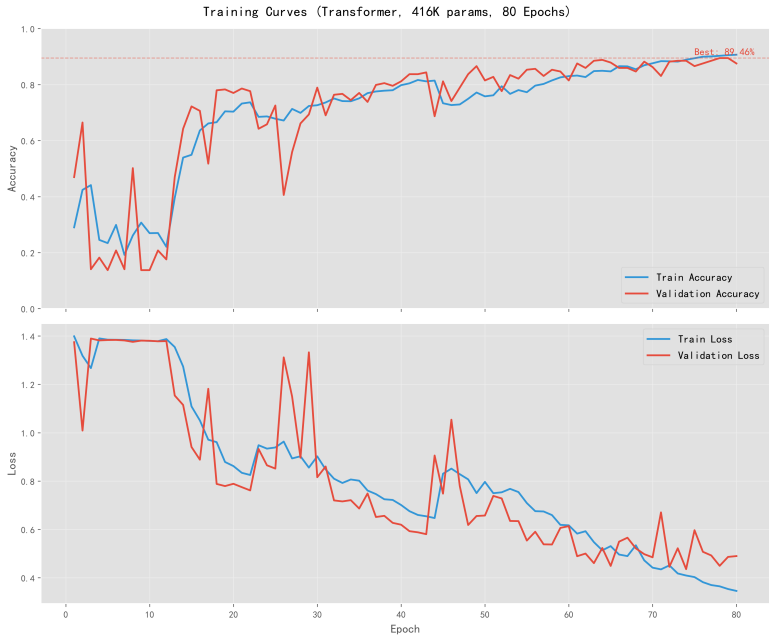


Figure 4. Loss and accuracy curves during training (picture credit: original)

DOI: 10.54254/2755-2721/2026.BA37518

The loss curve and accuracy curve during training are shown in Figure 4.
The model performance is summarized as shown in Table 2.

Table 2. Summary of model performance

Metric	Value
Number of parameters	416,068
Best validation accuracy	89.46%
Test accuracy	86.91%
Macro average F1	83.0%
Weighted average F1	86.9%

4.4. Confusion matrix analysis

The test-set confusion matrix intuitively presents the correct recognition and misclassification of each class, and clarifies the main misclassification patterns such as between smashes and drops. The confusion matrix is shown in detail in Figure 5.

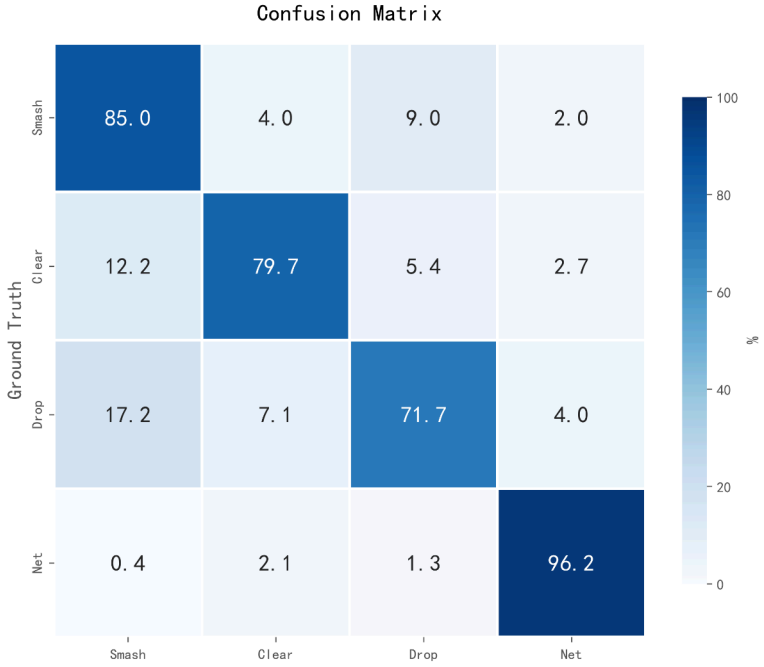


Figure 5. Test-set confusion matrix (picture credit: original)

Net shots exhibit the most favorable performance, as illustrated in this figure.: 230 of their samples are recognized correctly (96.2%) and only 9 samples are inaccurate. Because net shots almost always happen in the forecourt and look quite different from other strokes, their position gives the model a strong cue.

Smashes reach 85.0% recall but only 75.9% precision. The reason is that clears and drops look a lot like smashes during the take-off, since the racket wind-up and the rising center of gravity trace very similar skeleton paths in the early frames.

DOI: 10.54254/2755-2721/2026.BA37518

Clear achieves a stable F1 of 79.2%. Among 74 samples, 59 are correctly classified, and the misclassifications are mainly concentrated in confusion with smashes.

Drop has the lowest recall (71.7%), with 17 samples misclassified as smashes. The preparation poses of the two actions are almost identical, and distinguishing them by skeleton keypoints alone faces the problem of insufficient information.

From the 3.9-percentage-point gap between weighted F1 (86.9%) and macro-average F1 (83.0%), it can be seen that net shots, as the majority class, raise the weighted value. The impact of class imbalance still needs attention.

4.5. Per-class metrics

The F1 score, computed as the harmonic mean of precision and recall, is a standard metric for judging how well a classifier performs overall:

$$F_1 = \frac{2PR}{(P+R)} \tag{4}$$

Its value lies between 0 and 1, and it moves closer to 1 as the model improves. Unlike a simple average, the harmonic mean punishes large gaps between precision and recall, so a high F1 requires both to be decent at the same time. That makes it a fair indicator of overall discriminative power, especially when the classes are uneven.

The precision, recall and F1 score of each class are shown in Table 3.

Table 3. Per-class classification metrics

Class	Precision	Recall	F1 Score	Samples
Smash	75.9%	85.0%	80.2%	100
Clear	78.7%	79.7%	79.2%	74
Drop	81.6%	71.7%	76.3%	99
Net	96.6%	96.2%	96.4%	239
Macro average	83.2%	83.2%	83.0%	512
Weighted average	87.1%	86.9%	86.9%	512

4.6. Analysis of per-class performance

With F1 score of 96.4%, net shots substantially outperform all other classes. Spatially, the forecourt position alone is highly distinctive, and in terms of movement, short, small-range actions separate cleanly from strokes that involve large motion.

Smash detection has good sensitivity, but low precision. Clears and drops severely overlap with smashes in take-off poses. The existence of feints in real matches adds another layer of uncertainty.

Clears and drops have close F1 scores (79.2% and 76.3%), and the misclassification originates from the natural overlap of action patterns. Under long-distance camera shots, the skeleton difference between low flat drops and shallow clears is extremely small.

DOI: 10.54254/2755-2721/2026.BA37518

4.7. Main misclassification patterns

The confusion between drops and smashes is the most prominent weakness of the current model. 17 drops are judged as smashes, and 9 smashes are judged as drops. The two are consistent in take-off and racket preparation; the real difference lies in the power intensity at the moment of impact and the racket face angle, which skeleton data is inherently insufficient to depict.

Net shots are rarely misjudged (only 9 errors among 239 samples). The spatial feature constraint and short-range action patterns jointly provide sufficient discriminative evidence. This also validates the practical effect of court coordinate features in distinguishing forecourt and backcourt shots.

5. Discussion

5.1. Limitations

This method still has a few clear limitations worth pointing out. In terms of data, it is based on only five professional matches, so the sample size and scenario diversity are both limited. Professional players' actions are highly standardized, but amateur players vary much more. The generalization ability of the model on a wider population needs further verification.

At the feature level, only keypoint coordinates and speed are introduced. Visual cues such as racket trajectory and shuttlecock head direction have not been included. This information substantially helps distinguish smashes and drops, because the difference between the two lies exactly in the racket face angle and hitting power.

From an implementation standpoint, the current system operates entirely offline, as a real-time counterpart has not yet been developed. The 416K parameter count ensures compatibility with edge devices; however, empirical latency and memory usage remain to be measured on target hardware.

5.2. Future work

In future this project would be planned in two directions: adding auxiliary signals such as racket, and trying multimodal fusion to improve the network itself. Expanding the data scale and population coverage, as well as completing benchmark tests for real-time deployment, are also necessary steps to push the method toward practical use.

6. Conclusion

This paper presented a lightweight Transformer that classifies badminton strokes from pose keypoints. Starting with the 17 skeleton keypoints given by YOLO-Pose, the method fuses court mapping with movement speed into temporal features; a three-layer encoder of 416K parameters then performs the four-way stroke classification automatically. On ShuttleSet, the validation accuracy is 89.5%, the test accuracy is 86.9%, and the net-shot F1 reaches 96.4%. The results show that coarse-grained stroke recognition can be completed relying only on skeleton information, without ball tracking or racket detection modules.

This paper contributes mainly in two ways. One is an end-to-end pipeline that goes from raw match video straight to stroke classification, confirming that skeletal keypoints are enough for badminton action recognition. The other is the compact 416K-parameter design, which cuts computation noticeably without sacrificing accuracy and therefore suits edge deployment.

DOI: 10.54254/2755-2721/2026.BA37518

References

[1] Thomas, G., Gade, R., Moeslund, T. B., et al. (2017). Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding, 159*, 3–18.

[2] yo-WASSUP. (2026). Good-badminton. https://github.com/yo-WASSUP/Good-Badminton.

[3] Wang, W. Y., Shuai, H. H., Chang, K. S., et al. (2023). ShuttleSet: A human-annotated stroke-level singles dataset for badminton tactical analysis. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5126–5136.

[4] Zheng, C., Wu, W., Chen, C., et al. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys, 56*(1), 1–37.

[5] Ultralytics. (2025). YOLO-Pose: Real-time pose estimation. https://docs.ultralytics.com/tasks/pose/.

[6] Jiang, T., Lu, P., Zhang, L., et al. (2023). RTMPose: Real-time multi-person pose estimation based on MMPose. arXiv:2303.07399.

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

[8] Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 7444–7452.

[9] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.

[10] Shi, L., Zhang, Y., Cheng, J., et al. (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12026–12035.

[11] Wang, W. Y., Shuai, H. H., Chang, K. S., et al. (2022). ShuttleNet: Position-aware fusion of rally progress and player styles for stroke forecasting in badminton. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*(4), 4219–4227.

[12] Li, Q., Chiu, T. C., Huang, H. W., et al. (2024). VideoBadminton: A video dataset for badminton action recognition. *IEEE International Conference on Big Data (BigData)*, 1387–1392.