

TrackNetV3: Badminton Smash Measurer

Advisor: 藍崑站

Project Members: 楊家寶

Development Tools: Python, PyTorch, Dash, Google Colab

Test Environment: Google Colab (GPU acceleration), Windows/Linux

I. Introduction:

This project aims to develop an intelligent system, TrackNetV3, capable of accurately predicting and refining the shuttlecock trajectory in badminton match videos. The system consists of two core modules: the trajectory prediction module and the trajectory rectification module. The prediction module utilizes background estimation and data augmentation techniques (such as mixup) to remain robust even under visual interference. The rectification module addresses occlusion-induced gaps by refining predictions using inpainting techniques.

TrackNetV3 outperforms YOLOv7 and TrackNetV2 on the Shuttlecock Trajectory Dataset, achieving top scores in Accuracy, Precision, Recall, and F1-score.

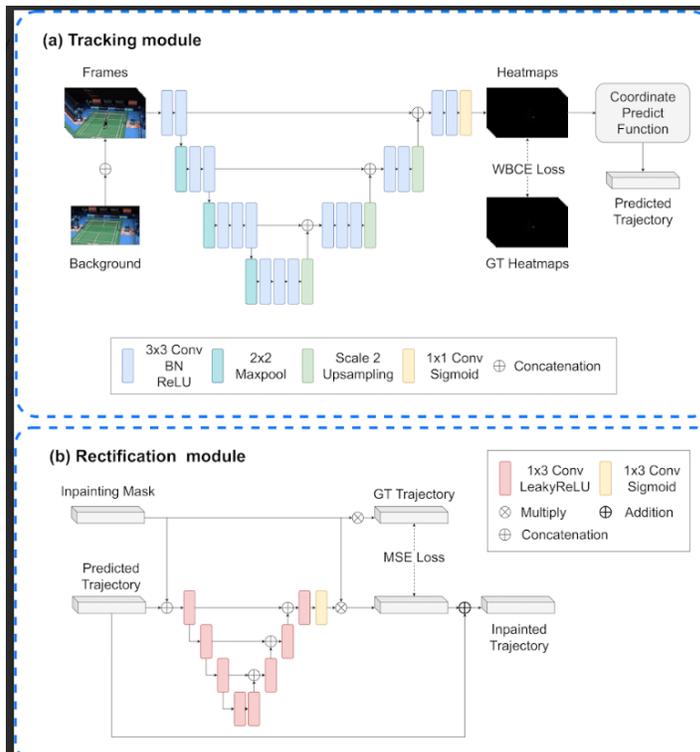
Model	Accuracy	Precision	Recall	F1	FPS
YOLOv7	57.82%	78.53%	59.96%	68.00%	34.77
TrackNetV2	94.98%	99.64%	94.56%	97.03%	27.70
TrackNetV3	97.51%	97.79%	99.33%	98.56%	25.11

System Architecture Diagram:

The TrackNetV3 architecture is composed of the following sequential stages:

1. Input Frames: Video frames are extracted and resized. Median background images are computed if necessary.
2. Trajectory Prediction:

- Frames are passed through the TrackNet model.
 - Output heatmaps represent the shuttlecock location per frame.
 - Heatmaps are post-processed to derive coordinates (x, y).
3. Temporal Ensemble (Optional):
- If enabled, overlapping frame predictions are combined using weight-based or average-based ensembling.
4. Trajectory Rectification:
- The inpaint module detects occluded or low-confidence predictions.
 - Refines predicted coordinates by inpainting based on visible context or past/future frames.
5. Speed Calculation:
- The system estimates real-world shuttlecock speed by measuring frame-to-frame displacement in pixels and converting it to meters using a calibrated pixel-to-meter ratio.
 - Speed is calculated only between frames where the shuttlecock is visible.
 - The distance in pixels is computed using the Euclidean distance formula between predicted (x, y) coordinates.
 - The real-world speed is then computed as:
Speed (m/s) = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times \text{pixel-to-meter ratio} / (1 / \text{frame rate})$
 - Finally, speeds are converted to km/h by multiplying the result in m/s by 3.6.
6. Output Generation:
- Annotated CSV files are produced, containing frame-wise predictions and speed.
 - Annotated video with real-time visual tracking and speed display is optionally generated.



System Components Overview:

- Trajectory Prediction Module: Inputs sequences of video frames and outputs heatmaps or coordinates.
- Inpaint Module: Analyzes predictions to detect occluded points and applies interpolation or neural network-based inpainting.
- Visualization Module: Generates annotated videos and provides a Dash-based interface for interactive error analysis.

II. Test Results:

The system was evaluated across the following aspects:

1. Model Performance: TrackNetV3 achieved 97.51% accuracy, 97.79% precision, 99.33% recall, and a 98.56% F1 score on the test dataset.
2. Visualization Results: The system outputs videos with annotated predicted trajectories, allowing visual comparison with ground truth.
3. Error Analysis Interface: An interactive Dash interface allows users to browse prediction discrepancies across rallies, classified as TP, FP1, FP2, FN, or TN.

[Insert Comparison.png and ErrorAnalysisUI.png here]

