A Project Report On

SMART BADMINTON PLAYER ANALYSIS AND PERFORMANCE OPTIMIZATION SYSTEM

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

By

Adrian Gilbert T: 21106015

Sanjita Shukla: 21106056

Ankit Yadav: 21106005

Under the Guidance of

Dr. Jaya Gupta

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

A.P. SHAH INSTITUTE OF TECHNOLOGY, THANE

UNIVERSITY OF MUMBAI

2024-2025

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "Smart Badminton Player Analysis and Performance Optimization System" is a bonafide work of **Adrian Gilbert T. (21106015), Sanjita Shukla (21106056) and Ankit Yadav (21106005)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning**).

Dr. Jaya Gupta
Guide

Prof. Sayali Sonje
Project Co-ordinator

Dr. Jaya Gupta
Head of Department

Dr. Uttam Kolekar
Principal

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval for B.E.

This project report entitled ***Smart Badminton Player Analysis and Performance Optimization System*** by ***Adrian Gilbert T, Sanjita Shukla, Ankit Yadav*** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science & Engineering (Artificial Intelligence & Machine Learning)***, ***2024-25***.

Examiner Name	Signature
1.________________	______________
2.__________________	_______________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Adrian Gilbert T. 21106015)

(Sanjita Shukla 21106056)

(Ankit Yadav 21106005)

Date:

Abstract

Badminton is a high-speed sport that requires agility, accuracy, and strategic placement of shots. While current performance analysis systems are mostly quantitative in nature, focusing on speed and number of shots, qualitative information that can assist players in improving their technique is lacking. This paper introduces a Smart Badminton Player Analysis and Performance Optimization System, an artificial intelligence-based system that examines player movement, stroke production, and playing patterns through computer vision and deep learning algorithms. The system combines several detection models, such as YOLOv8 for player detection, TrackNet V3 for shuttlecock tracking, CNN for filtering non-play frames, and Roboflow Pose Model for stroke classification. Court keypoints are manually chosen and saved to maintain accurate tracking of movements in the playing court. Through the integration of these methods, the system separates key moments of gameplay, detects shot types, and offers data-driven performance analysis. We successfully addressed the key challenge of integrating multiple models into a seamless pipeline, achieving accurate game footage analysis allowing coaches and amateur players to improve their gameplay by reviewing the output footage. Future development will focus on increasing stroke classification, adding real-time coaching feedback, and incorporating sensor-based tracking. This work sets the stage for a complete, AI-augmented badminton coaching system that closes the loop between raw performance data and actionable skill enhancement.

Keywords

Badminton analytics, deep learning, player detection, shuttlecock tracking, pose estimation, stroke classification, AI in sports, game footage analysis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

AI	*Artificial Intelligence*
ML	*Machine Learning*
CNN	*Convolutional Neural Network*
YOLO	*You Only Look Once*
BWF	*Badminton World Federation*
mAP	*Mean Average Precision*
IoU	*Intersection over Union*
CIoU	*Complete Intersection over Union*
MSE	*Mean Squared Error*
TPU	*Tensor Processing Unit*
ONNX	*Open Neural Network Exchange*
CUDA	*Compute Unified Device Architecture*
TPUs	*Tensor Processing Units*
ResNet	*Residual Network*
CSV	*Comma-Separated Values*
ReLU	*Rectified Linear Unit*
CSPDarkNet53	*Cross-Stage Partial Darknet-53*
AdamW	*Adaptive Moment Estimation with Weight Decay*
RMSprop	*Root Mean Square Propagation*
SSD	*Single Shot MultiBox Detector*
GPU	*Graphics Processing Unit*
DFL loss	*Distribution Focal Loss*
GNNs	*Graph Neural Networks*

Chapter 1

Introduction

Badminton is a highly dynamic sport requiring rapid reflexes, precise shot-making, and strategic movement. Current analysis tools primarily focus on quantitative statistics such as rally duration and shuttle speed, overlooking qualitative aspects like shot selection, stroke efficiency, and player movement patterns.

This project introduces a Smart Badminton Player Analysis System, leveraging deep learning and computer vision techniques to deliver detailed player performance evaluation. Unlike traditional systems that focus on isolated elements such as shuttle tracking or player detection, this system integrates multiple AI models into a single pipeline for holistic match analysis.

Key Contributions:

- Automated Video Processing: Extracts meaningful gameplay moments from match footage.
- Multi-Model Integration: Combines YOLOv8, TrackNet V3, and pose estimation models.
- Overlay-Based Visualization: Displays statistics directly onto gameplay footage.
- Real-Time Analytics Potential: Lays groundwork for future real-time coaching applications.

The system follows a step-by-step approach where each phase builds upon the previous one:

1. Load Court Keypoints & Overlay on Video
2. Detect Players & Overlay on Video
3. Detect Shuttlecock & Overlay on Video
4. Extract Hit Frames
5. Predict Poses from Hit Frames & Overlay on Video
6. Generate Statistical Graphs & Overlay on Video
7. Final Output: Processed Video with All Overlays

This structured methodology ensures comprehensive and reliable performance analysis.

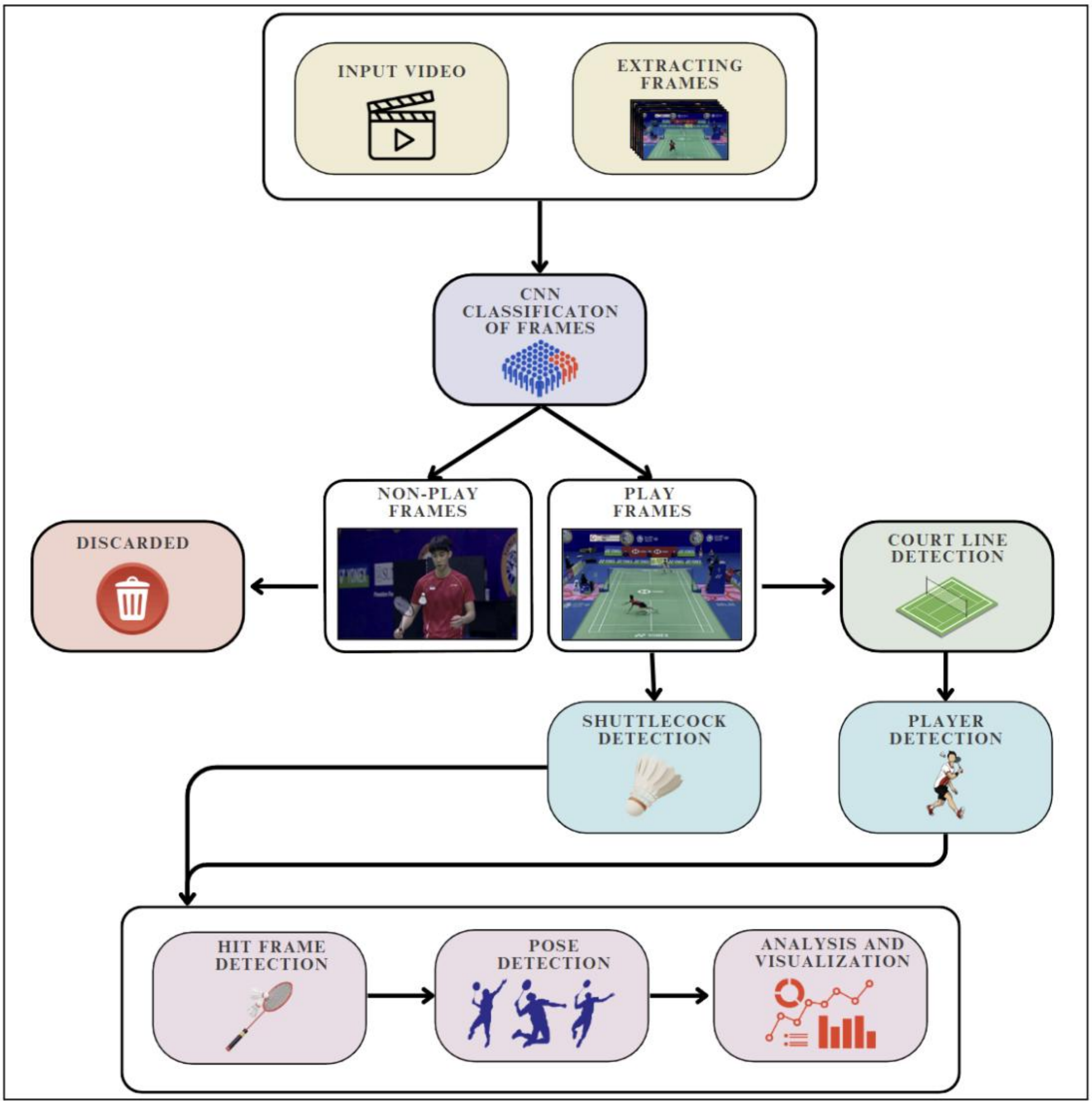

Fig 1. Workflow of the Proposed System

Chapter 2

Literature Survey/ Existing system

Karpathy et al. [1] presented Convolutional Neural Networks (CNNs) for large-scale video classification, establishing a platform for automated sports analytics. Their research proved CNNs' suitability in processing dynamic visual information, an essential aspect of badminton analytics. The research highlighted how deep learning models can handle enormous video data, offering real-time information on player movement and shot types. King et al. [2] also explored the biomechanics of shuttlecock speed in jump smashes. They discovered robust correlations between racket velocity and shuttlecock speed, serving as a foundation for deep learning models to forecast shuttle trajectories and improve training regimes. Through motion capture data analysis, they discovered the major biomechanical parameters that have an impact on stroke efficiency, which can be incorporated into AI-driven badminton player training systems. Upulie et al. [3] reviewed real-time object detection using the YOLO framework, highlighting its application in sports analytics. Their findings emphasized YOLO’s superior speed and accuracy, making it ideal for real-time shuttlecock and player tracking in badminton. They explored how YOLO’s real-time processing capabilities enable instant feedback, allowing athletes to adjust their playing techniques during training. Huang et al. [4] created TrackNet, a high-speed and small-object tracking-specialized deep learning model. TrackNet's strong performance in detecting shuttlecock movement even under occlusion greatly contributes to automated badminton analysis. Their study also emphasized how the model can improve training approaches by offering accurate trajectory predictions for various shot types. Szczuko et al. [5] investigated deep neural networks for human pose estimation from low-resolution depth images. Their work benefited pose recognition methods used in badminton stroke classification and movement evaluation. They showed how enhanced pose estimation would enable coaches to evaluate players' stances and movements, and detect areas for technique and agility improvement. Howard et al. [6] proposed MobileNets, a light-weight CNN model designed for mobile platforms. The low computational expense of the model allows for real-time inference, making it an ideal candidate for mobile-based badminton performance analysis. Their work demonstrated that MobileNets was capable of giving real-time feedback in mobile coaching apps, thus making AI-driven coaching feasible for amateur players.

Zhang et al. [7] introduced SADAM, a variant of the Adam optimizer. Their research enhanced convergence rate and accuracy, which would be advantageous for deep learning models in sports, especially player tracking and stroke classification. They emphasized how gradient-based learning optimization could improve the stability and performance of badminton analytics models. Neil et al. [8] investigated Transformers in computer vision for image recognition, where better feature extraction and classification accuracy were achieved. This work paved the way for badminton analytics with Transformer-based architectures for increased performance insight. They illustrated how self-attention mechanisms enhance the context-specific comprehension of gameplay videos, improving player movement analysis. Liu et al. [9] presented SSD (Single Shot MultiBox Detector), an object detection framework in real-time. Their research formed the basis of shuttlecock and player detection models for badminton based on deep learning methods. They offered a comparison with conventional detection models, where they proved SSD's effectiveness in high-speed sports settings. Gavrilescu et al. [10] optimized Faster R-CNN for real-time object detection for latency-constrained applications. Their results aided in object detection pipeline optimization badminton video analytics. They explored how deep feature pyramid networks in Faster R-CNN enhance shuttlecock tracking accuracy. He et al. [11] created Masked Autoencoders (MAE) for large-scale vision tasks. Their self-supervised learning approach improved model robustness, which can be used in sports video analysis for automatic event segmentation. Their work proposed how MAE-based models could assist in classifying various types of strokes in badminton with little supervision. Li et al. [12] compared Vision Transformers (ViTs) on detection transfer learning. Their paper highlighted the use of ViTs for badminton analytics by enhancing object detection performance on a variety of datasets. They illustrated how fine-tuning ViTs on sports datasets could provide very accurate tracking models for shuttlecock and player movements.

Bhatti et al. [13] summarized Graph Convolutional Networks (GCNs) and their usage in computational intelligence. Their work gave an overview of the use of GCNs in multi-object tracking and motion analysis in sports. They explored how GCNs facilitate real-time coordination tracking between players in doubles badminton games. Touvron et al. [14] analyzed deeper image transformers to enhance classification performance. Their results indicate that transformer-based models can enrich badminton analytics by offering more precise shot classification and strategy forecasting. They illustrated how transformer models are superior to traditional CNNs in detecting subtle differences in stroke execution. Zhao et al. [15] suggested recurrent layer aggregation in deep CNNs to boost feature retention. The method aids badminton stroke recognition with accurate classification of different shot types. Their research also investigated the role of recurrent feature aggregation in improving the interpretability of motion sequences in sports analytics. Lin et al. [16] proposed Feature Pyramid Networks (FPN), which has greatly enhanced multi-scale object detection. This benefits shuttlecock detection in badminton videos, where shuttlecock size changes from frame to frame. They illustrated how FPN-augmented YOLO models outperform baseline object detection methods in badminton match analysis.

Zou et al. [17] improved human pose estimation with stacked hourglass networks. Their system enhanced joint localization accuracy to benefit player movement analysis and stroke classification in badminton. They highlighted how stacked hourglass networks assist in enhancing player agility evaluation by monitoring footwork patterns. Liu et al. [18] proposed a Deep Dual Consecutive Network (DDCN) for pose estimation. They optimized feature extraction and tracking in sports use cases, specifically in real-time badminton analysis. They emphasized DDCN's dominance in pose stability across extended badminton rallies. Fang et al. [19] proposed AlphaPose, a state-of-the-art multi-person pose estimation system. Their work advances badminton performance analysis by allowing accurate tracking of player postures in rallies. They showed that AlphaPose was able to differentiate between aggressive and defensive stances in professional games. Yang et al. [20] presented Graph Neural Networks (GNNs) for human pose tracking and estimation. Their work benefits badminton strategy evaluation through enabling multi-player movement analysis as well as studies on team coordination. They presented how GNN-based models may derive spatial relations among players during real-time tactical analysis. Chen et al. [21] established TrackNetV3 with augmented shuttlecock tracking via deep learning-based trajectory rectification. The study deployed the latest data augmentation methods to polish shuttlecock movement prediction and avoid false alarms when tracking fast-speed badminton play. Beal et al. [22] researched AI-based game strategy optimization through reinforcement learning techniques. They showed how strategic decision models can assist sportsmen in adjusting to changing game situations, making them more responsive and efficient in competitive badminton games.

Cui et al. [23] introduced a Siamese network-based tracking system that improves motion tracking robustness in sports use. Their system effectively suppresses drift for long gameplay sequences, enabling more stable and accurate tracking of badminton players during a match. Bochkovskiy et al. [24] presented YOLOv4, an efficient real-time object detection model, with enhanced speed and accuracy. The model has gained extensive use in badminton analytics, especially in following shuttlecock trajectories and capturing player locations. Chen et al. [25] analyzed badminton players' behavior in telecast videos, utilizing AI-driven analysis methods to reveal tactical movement patterns. The study presented new knowledge on decision-making patterns and performance trends of professional badminton players. Chen et al. [26] introduced an adaptive object tracking system to suit sports analytics. Their approach enhanced stroke recognition accuracy with the incorporation of multi-scale feature extraction, which optimized real-time tracking of shuttlecock and players. Khanam et al. [27] explored YOLOv5's architectural improvements, outlining its use in high-speed motion analysis and shuttlecock tracking for the badminton sport. Bakirci et al. [28] compared YOLOv8's object detection accuracy boost and real-time inference, which proved beneficial in monitoring shuttlecock movement in high-intensity sports. Li et al. [29] implemented YOLOv8 in UAV-based sports video analysis and offered sophisticated knowledge of badminton gameplay from overhead views. Hussain et al. [30] investigated AI-based defect detection strategies, illustrating how computer vision methods are usable in reengineering shuttlecock trajectory analysis and real-time motion monitoring in badminton.

Chapter 3

Limitation of Existing system

3.1 Research Gaps and the Need for an Integrated Approach

Despite significant advancements in sports analytics, badminton remains an underdeveloped domain in terms of automated performance analysis. Unlike mainstream sports such as football and basketball, where extensive machine learning-driven analytics exist, badminton players lack access to cost-effective, AI-powered solutions. Current methodologies suffer from several limitations, making it difficult for players and coaches to extract meaningful insights from match footage.

3.2 Limitations of Existing Systems

1. High Cost of Implementation

Most existing AI-driven sports analytics platforms are tailored for high-budget professional teams and require specialized hardware and proprietary software. These solutions are neither affordable nor accessible to amateur and semi-professional players. Implementing machine learning-based tracking often requires expensive GPU-powered systems, high-resolution cameras, and customized software licenses, limiting its use at grassroots levels.

2. Immense Data Requirement

Deep learning models thrive on large-scale annotated datasets. However, badminton lacks publicly available datasets that comprehensively include player movements, shuttlecock tracking, and stroke classification. The absence of a standardized training dataset restricts the development of robust AI models and forces researchers to rely on limited, manually labelled data, reducing system accuracy and generalizability.

3. Lack of an All-in-One System

Most existing research focuses on isolated aspects of badminton analytics, such as court line detection, shuttlecock tracking, or player pose estimation. However, there is no single unified system that combines all these functionalities into a comprehensive tool for match analysis. Players must rely on multiple fragmented solutions, making the workflow cumbersome and inefficient.

4. Limited Availability of Open-Source Solutions

Many state-of-the-art sports analytics platforms are proprietary, making them inaccessible to independent researchers and non-professional athletes. The lack of open-source frameworks limits innovation, making it difficult for new researchers to build upon existing models without significant investment.

5. Lack of Standardization in Metrics and Evaluation

Unlike other sports where metrics such as pass accuracy (football) or shooting efficiency (basketball) are well-defined, badminton lacks standardized performance metrics. Current studies often propose custom evaluation techniques, making it difficult to compare findings or create a benchmark system for stroke analysis, player positioning, and shuttle movement prediction.

6. Real-Time Processing Limitations

Existing solutions for badminton match analysis often require offline processing due to the computational complexity of deep learning models. Real-time analytics is still a challenge because of the need for high-end hardware, which is inaccessible to most players and small-scale training academies.

Chapter 4

Problem Statement and Objective

Analyzing badminton matches manually is time-consuming and lacks accuracy due to human subjectivity and observational biases. Coaches and players often rely on manual video reviews, which are inefficient and fail to provide detailed, objective insights into player performance. Existing AI-based sports analysis tools focus on specific aspects such as shuttle tracking, court detection, or player tracking, but there is no comprehensive system that automates the entire analysis process in a lightweight, user-friendly manner.

Current solutions require multiple separate models, making implementation complex and computationally expensive. Additionally, the lack of an all-in-one system integrating court keypoint detection, player and shuttle tracking, stroke classification, and performance analytics prevents seamless match analysis. To address these challenges, this project aims to develop a fully automated, AI-powered badminton match analysis system that streamlines performance tracking, minimizes manual effort, and enhances training effectiveness.

4.1 How This Project Overcomes These Challenges?

Our Smart Badminton Player Analysis System addresses these limitations by offering a cost-effective, unified, and automated approach to badminton performance analysis. By leveraging:

1. Deep learning techniques such as YOLOv8, TrackNet V3, and Roboflow Pose Models, our system enables accurate player and shuttlecock tracking.
2. Pre-trained models to reduce data dependency and enhance performance with limited labeled datasets.
3. Affordable, open-source computer vision solutions, making it accessible for players at all levels.
4. An all-in-one system that processes an entire match video, extracting court keypoints, detecting players, tracking shuttlecock movements, analyzing stroke types, and overlaying statistical data on the video.
5. Standardized analytics and visualization tools, offering a single, intuitive interface where users can upload a match video and receive a processed video with key insights.
6. Future integration of real-time processing, enabling immediate feedback during training sessions.

Developing an open-source dataset and analytics framework for badminton will further standardize research, making AI-driven badminton analytics accessible, scalable, and impactful for both professional and amateur players alike.

4.2 Objectives

The primary objective of this project is to develop an AI-powered system that processes badminton match footage and provides detailed performance insights. The system aims to:

1. Automate match analysis by leveraging deep learning and computer vision techniques.
2. Filter unnecessary frames to focus only on key gameplay moments.
3. Detect court boundaries and keypoints to ensure accurate tracking.
4. Track player movements using object detection and tracking models.
5. Detect the shuttlecock's trajectory for shot analysis.
6. Identify and classify different badminton strokes using pose estimation models.

7. Overlay extracted insights onto the match footage, providing a visually enriched output.

8. Generate real-time performance statistics, helping players and coaches improve training strategies.

9. Ensure computational efficiency, making the system lightweight and accessible for different user levels.

10. Develop a standardized framework for badminton analytics, contributing to future research and development in AI-powered sports performance analysis.

By achieving these objectives, the project will offer a comprehensive, cost-effective, and automated solution for badminton performance evaluation, benefiting athletes, coaches, and researchers alike.

Chapter 5

Proposed System

The Smart Badminton Player Analysis System is designed as a fully automated AI-powered system that processes badminton match footage to provide comprehensive performance analysis. The system follows a brick-by-brick workflow, where outputs from each phase are saved and used for subsequent processing, ensuring computational efficiency and accurate results.

The workflow consists of six primary phases:

1. Non-Play Frame Detection
2. Court Keypoints Detection
3. Player Detection
4. Shuttlecock Detection
5. Hit Frame Detection
6. Pose Detection and Stroke Analysis

The final output is a video overlaying all the extracted information, giving players and coaches an intuitive way to analyze performance.

5.1 Non-Play Frame Detection

Objective

Filter out non-play elements such as slow-motion replays, highlights, and reaction shots to ensure that only pure gameplay footage is analyzed.

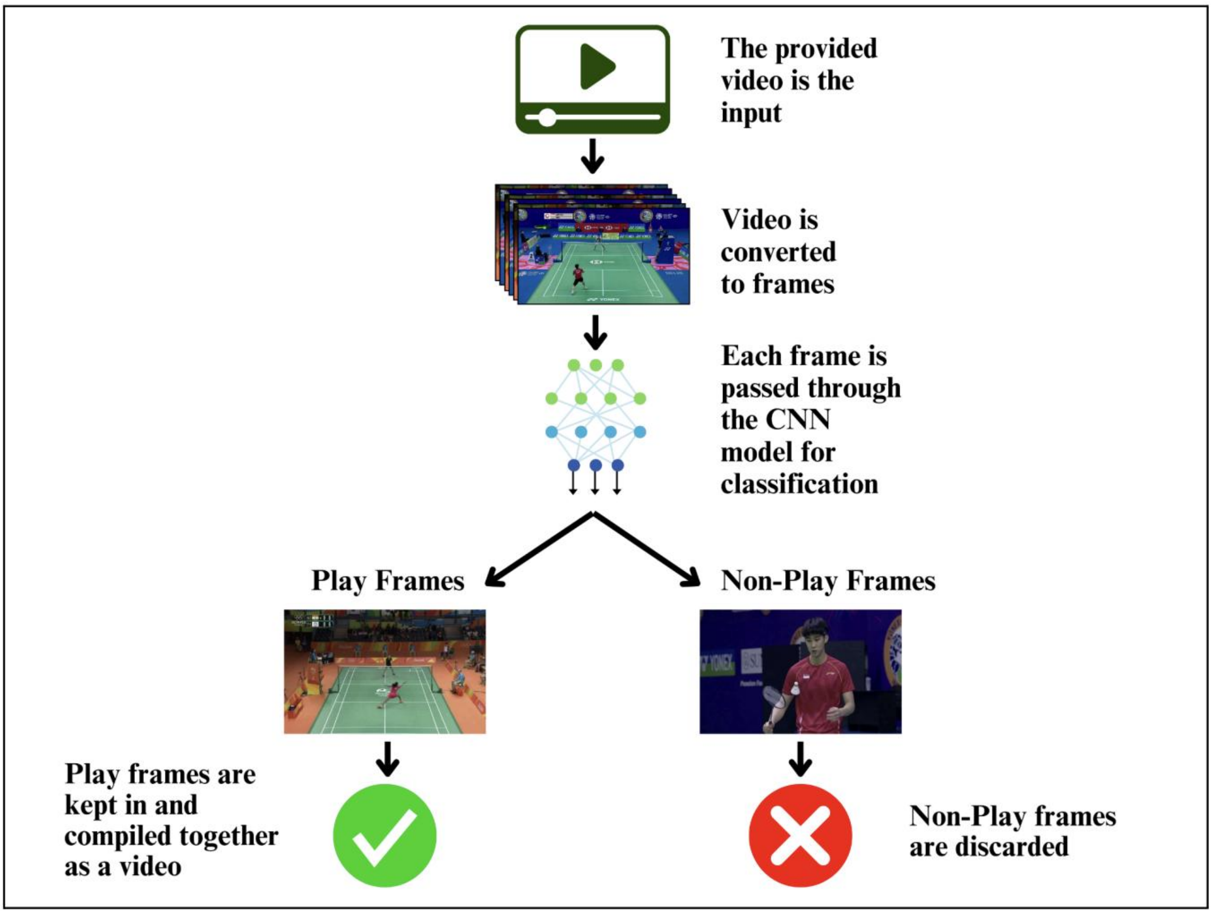

Fig 5.1 Workflow of the Play and Non-Play Frame CNN Classification Model System

Fig 5.2 Play Frame Samples

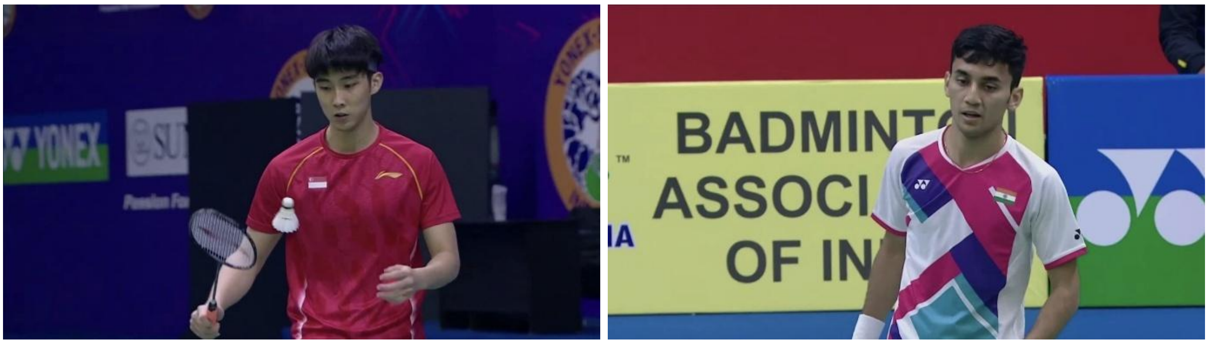

Fig 5.3 Non-Play Frame Samples

5.1.1 Process Overview

- The user provides 10 timestamps representing play frames and 10 timestamps for non-play frames.
- A Convolutional Neural Network (CNN) model is trained on manually labeled badminton match footage to classify frames as either play or non-play.
- The system removes non-play frames, ensuring that only relevant gameplay sequences are processed.
- If the user prefers, they can manually edit the footage beforehand or skip this step.

5.2 Court Keypoints Detection

Objective

Accurately identify and store court boundaries (key points) to aid in player movement tracking and shuttle trajectory analysis.

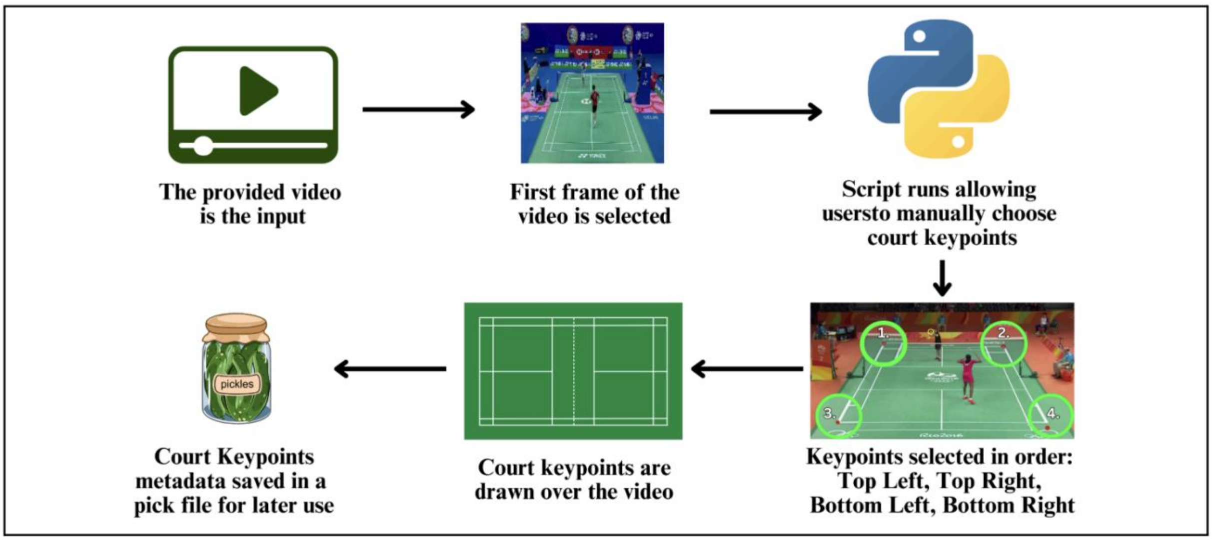

Fig 5.4 Workflow of Court Line Detection Strategy in our System

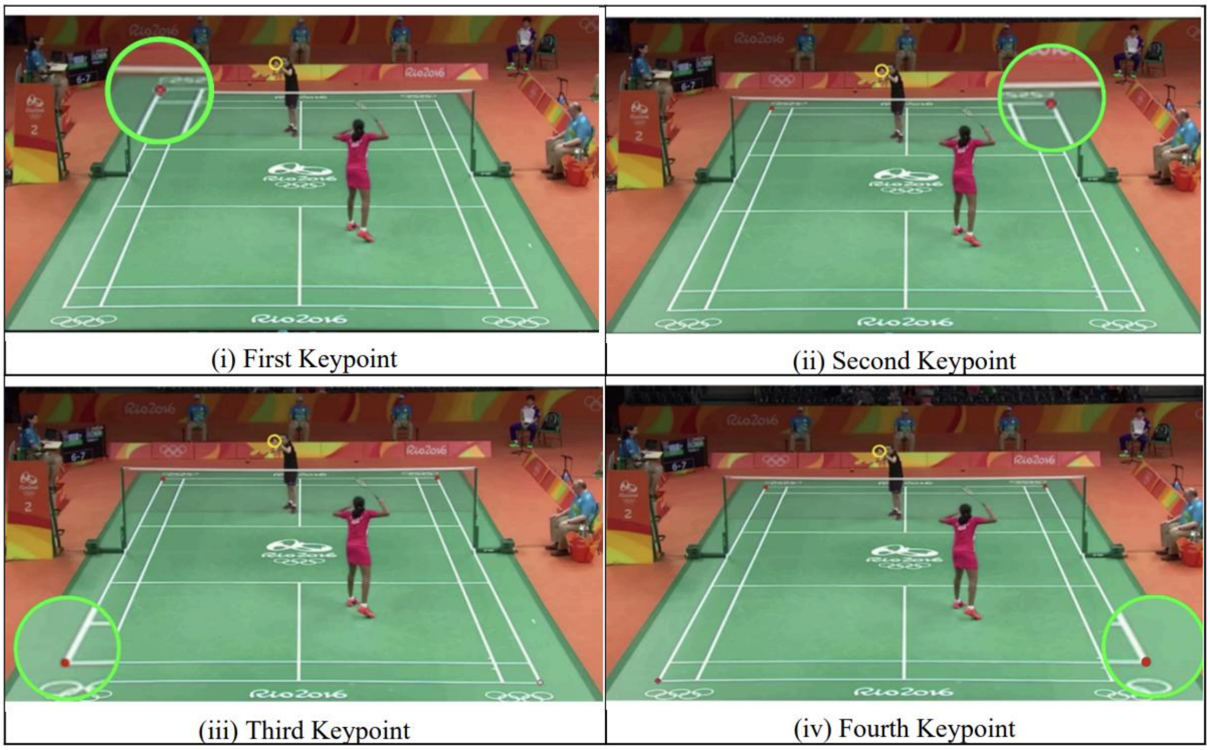

Fig 5.5 Court Keypoints (Boundaries)

5.2.1 Process Overview

- The user manually selects court keypoints from a video frame.
- These keypoints are stored using Python's pickle module for easy retrieval.
- Once saved, the system automatically loads the keypoints in future analyses to avoid repetitive manual input.
- OpenCV is used for facilitating manual selection and ensuring accurate court boundary detection.

5.3 Player Detection

Objective

Detect and track only the two players on the court, filtering out irrelevant objects like referees and audience members.

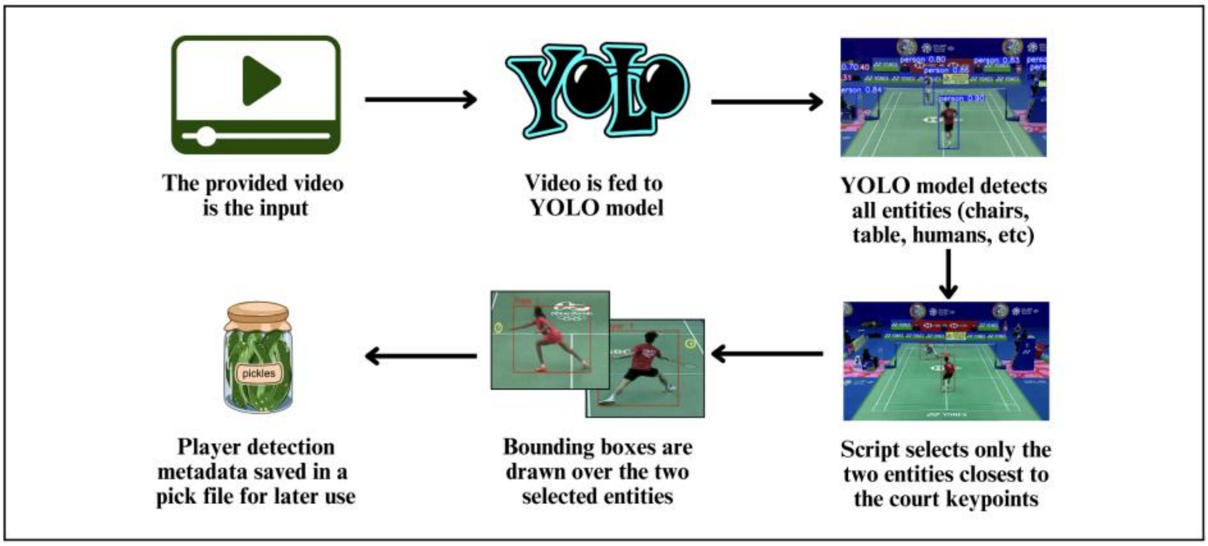

Fig 5.6 Workflow of Player Detection Model

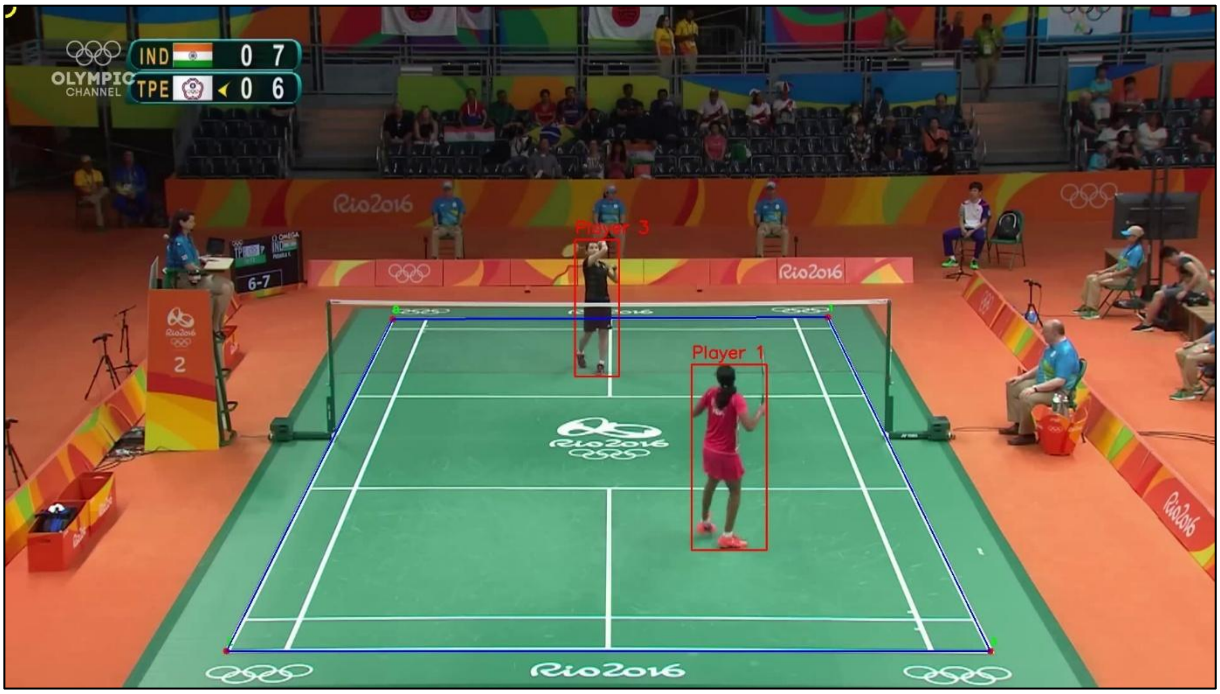

Fig 5.7 Detected Players on Court

5.3.1 Process Overview

- YOLOv8 was chosen due to its optimal speed-accuracy balance for real-time tracking.
- The model detects multiple objects in a frame, and a custom script filters out non-player objects, ensuring only the two players inside the court are retained.
- Bounding boxes (bboxes) are drawn around detected players and saved for subsequent tracking.
- Player detection data is stored in a pickle file to avoid redundant computations.

5.3.2 Why YOLOv8?

- Faster inference time than more advanced YOLO versions.
- Lower computational cost while maintaining high accuracy.
- Easily adaptable for real-time badminton player tracking.

5.4 Shuttlecock Detection

Objective

Accurately detect and track the shuttlecock, even in fast-motion frames, to analyze player strokes.

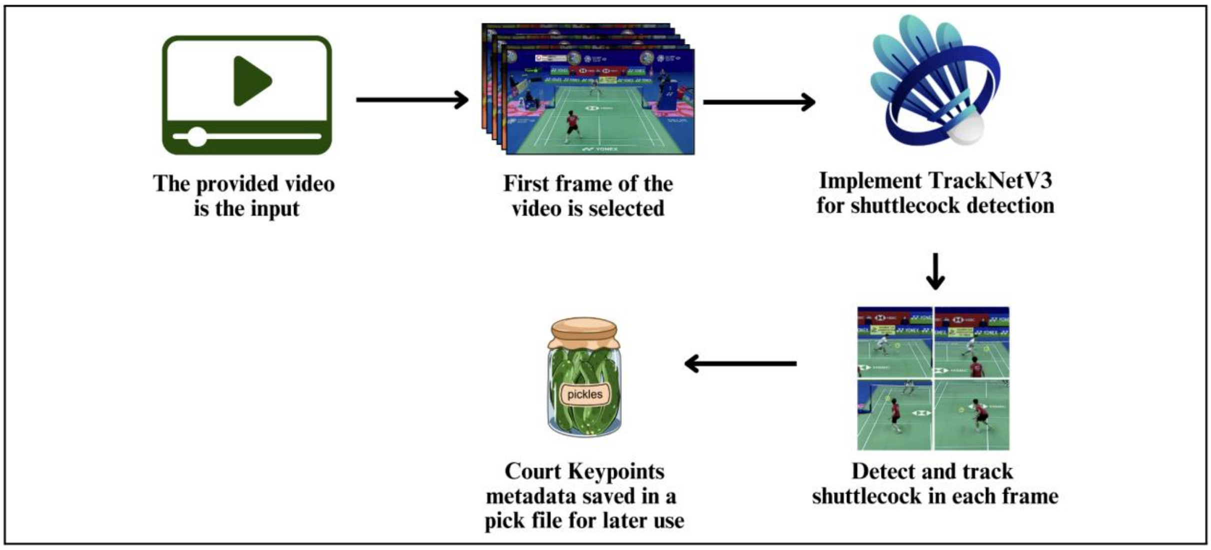

Fig 5.8 Workflow of the TrackNet V3 Model for Shuttlecock Tracking in our System

Fig 5.9 Shuttlecock Detection

5.4.1 Process Overview

- Classic object detection techniques (contour detection, template matching) were first used but failed in 20-30% of cases, especially during high-speed rallies.
- TrackNet V3 was implemented, a deep learning model designed for fast-moving sports objects.

- Since the original TrackNet V3 model is not readily available, its architecture was replicated from research papers.
- A custom dataset was created with manually labeled shuttlecock images from:
    - Olympic badminton matches
    - Amateur and professional match footage
    - Shuttlecocks in various environments
- Shuttlecock detection data is saved in a pickle file for use in later phases.

5.4.2 Why TrackNet V3?

- Designed specifically for high-speed object tracking in sports.
- Handles motion blur and rapid shuttle movement better than conventional methods.
- Ensures crucial frames are not missed, improving stroke analysis accuracy.

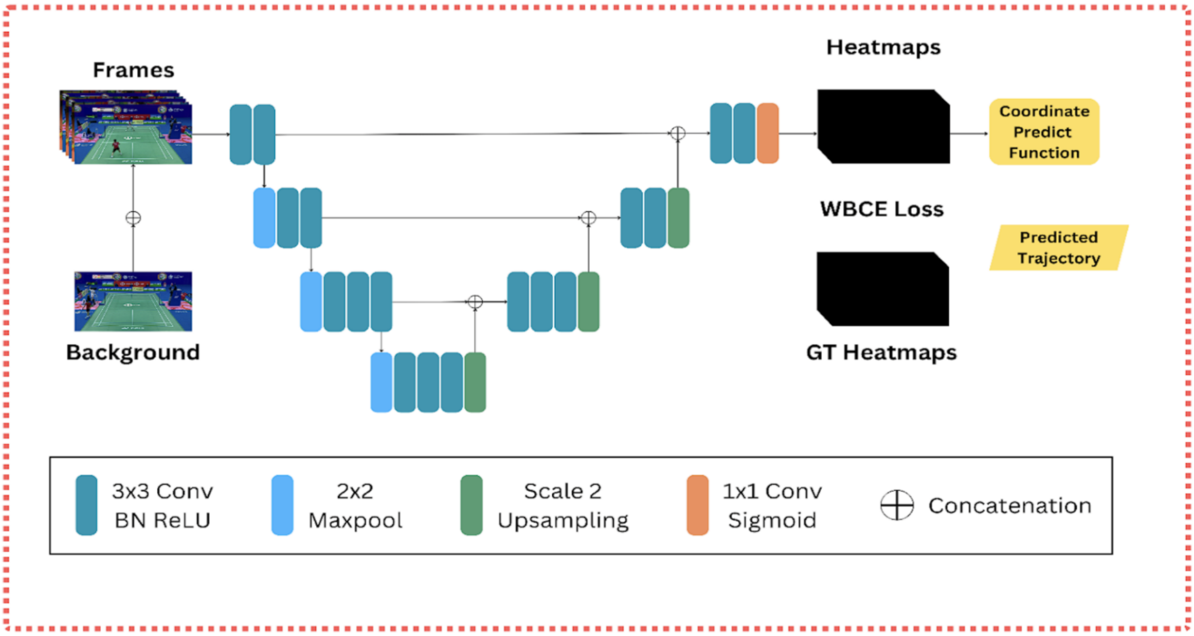

Fig 5.10 TrackNetV3 Model Architecture

Fig 5.10 demonstrates a flowchart showing the workflow of the TrackNet V3 model for shuttlecock detection in a badminton analysis system. It starts with video input, which is processed into frames. TrackNet V3 is used to detect the shuttlecock in each frame. The model continuously does detection and tracking, which guarantees precise localization of the shuttlecock. Lastly, the shuttlecock information detected is saved in a pickle file to be reused in the future. The flowchart illustrates the systematic method of shuttlecock tracking through the application of deep learning methods.

5.5 Hit Frame Detection

Objective

Identify the exact frame where a player makes contact with the shuttlecock, ensuring accurate input for pose detection.

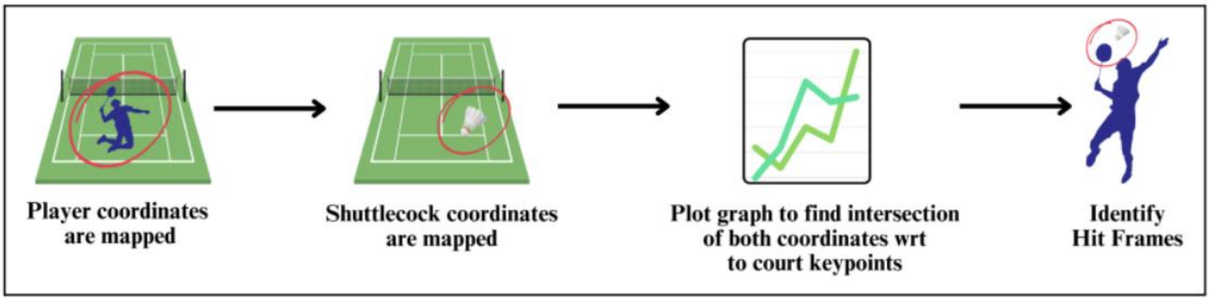

Fig 5.11 Workflow of Hit Detection System

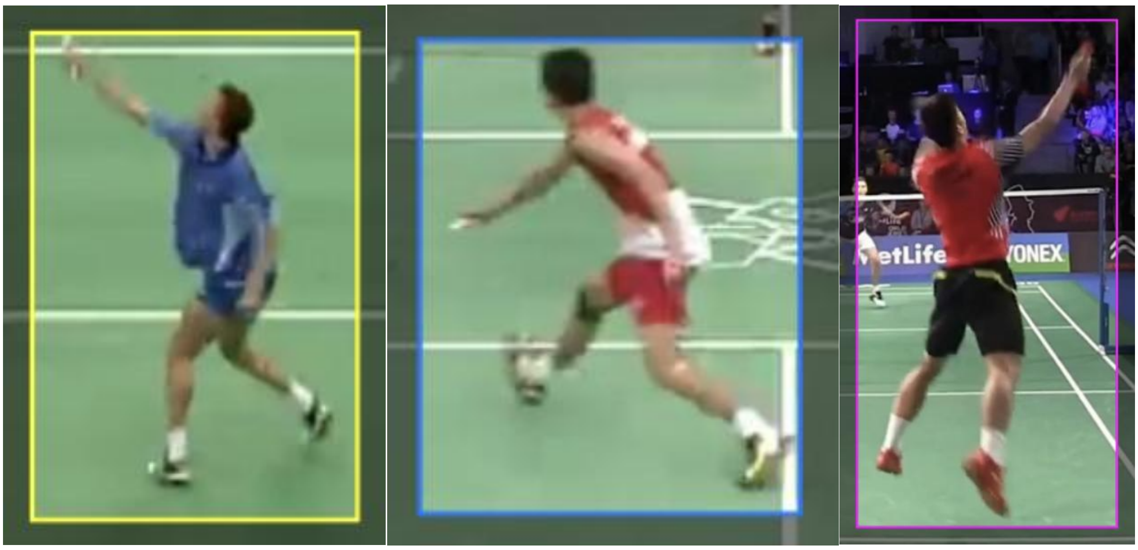

Fig 5.12 Hit Frame Detection Samples

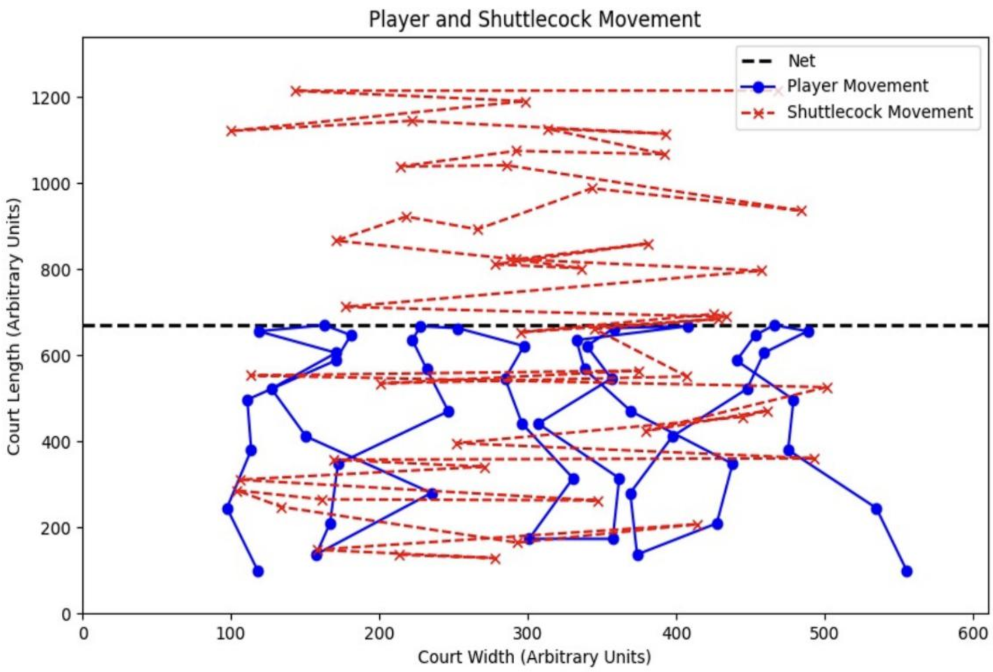

Fig 5.13 Player Movement w.r.t Shuttlecock Movement Fraph

The graph in Fig 5.12 visualizes player and shuttlecock movement on a badminton court, highlighting how their positions were used to identify hit frames—the moments when the player makes contact with the shuttlecock. The blue line with dots represents the player's movement across the court. The red dashed lines with 'x' markers indicate the shuttlecock's trajectory. The black dashed line represents the net, dividing the two halves of the court.

By analyzing the points where the player's movement closely aligns with the shuttlecock's changing trajectory, we can determine the hit frames—the specific frames where the shuttlecock is struck. These keyframes are crucial for pose estimation, as they allow us to analyze the player's body posture, stroke technique, and positioning at the moment of impact. Identifying these frames ensures that pose estimation focuses only on relevant moments rather than the entire video, improving accuracy and reducing unnecessary computations. This method enhances our ability to study player mechanics and refine performance analysis in badminton.

5.5.1 Process Overview

- TrackNet V3 generates a CSV file containing shuttlecock coordinates across all frames.
- The player's bounding box coordinates are retrieved from the Player Detection phase.
- The shuttlecock's position is cross-referenced with player's reach to detect the hit frame.
- Hit frame information is saved in a pickle file for use in later phases.

This step ensures precise input for Pose Detection, preventing errors caused by misclassified frames.

5.5.2 Why Use a Combination of TrackNet V3 & YOLOv8?

- YOLOv8 ensures accurate player tracking, while TrackNet V3 accurately tracks the shuttlecock.
- Their combination provides higher precision in detecting the hit frame.

5.6 Pose Detection and Stroke Analysis

Objective

Analyze the player's body posture at the hit frame to classify the type of stroke executed.

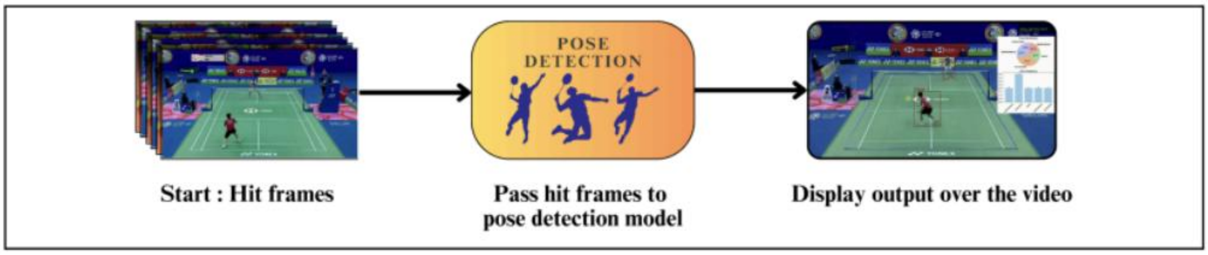

Fig 5.14 Workflow of Pose Detection model

Fig. 5.15 Examples of all the Poses in the System

A pretrained pose predictor model from Roboflow Universe is used for badminton-specific pose detection. The model detects key joint positions and classifies poses into 7 stroke types as show in Fig 5.14, which are as follows:

- Backhand Clear
- Backhand Lift
- Backhand Serve
- Forehand Clear
- Forehand Lift
- Ready Position
- Smash

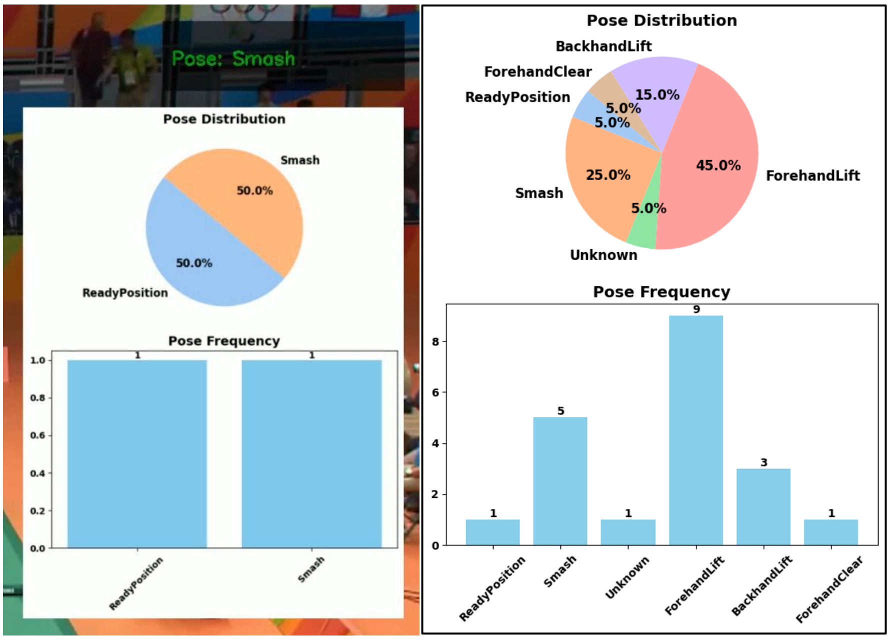

Fig 5.16 Combined Chart and Pose Statistics

5.6.1 Process Overview

Visual Overlays on Video Output:

- Bar Graph: Displays the frequency of different strokes detected.
- Pie Chart: Shows stroke type distribution throughout the game.
- Real-Time Pose Indicator: Displays the detected stroke in real-time at the top-right of the video.

5.6.2 Why Roboflow's Pose Model?

- Pretrained for badminton-specific pose classification.
- Saves time in developing a custom pose model from scratch.
- Ensures high accuracy in detecting and categorizing strokes.

5.7 Final Workflow Summary

The system processes a badminton match sequentially, leveraging each stage's output to inform the next, ensuring a streamlined and structured analysis pipeline. The core components of the system are as follows:

1. **Non-Play Frame Detection**: This stage filters out irrelevant footage such as player warm-ups, breaks, and pauses between rallies, ensuring only meaningful in-game sequences are analyzed. This significantly reduces computational overhead and focuses attention on performance-critical moments.

2. **Court Keypoint Detection**: The system detects and maps the court's boundaries and keypoints, including service lines and net positions. This spatial understanding is essential for accurate shuttlecock tracking, player positioning analysis, and determining shot legality.

3. **Player Detection**: Both players are identified and tracked throughout the match using computer vision techniques. The system ensures consistent player association (e.g., Player A and Player B) across frames, enabling reliable individual statistics and movement heatmaps.

4. **Shuttlecock Detection**: Fast and accurate shuttlecock tracking is performed, even during high-speed exchanges. This module captures the shuttlecock's trajectory, speed, and direction, which is crucial for rally reconstruction and stroke classification.

5. **Hit Frame Detection**: The system pinpoints exact frames where the shuttlecock is struck. These frames are critical for understanding shot timing, stroke type, and player reaction time. Identifying these moments allows for precise segmentation of rallies and shot sequences.

6. **Pose Detection & Analysis**: Advanced pose estimation algorithms are used to analyze player body movements. This data helps classify different types of strokes—such as smashes, drops, clears, and net shots—and enables biomechanical feedback. Additionally, overlays visualize shot technique, balance, and footwork.

The final output is a fully processed and annotated match video, enriched with statistical overlays such as shot distribution, rally lengths, player movement intensity, and stroke accuracy. These insights are invaluable for players aiming to refine their technique and for coaches developing tactical strategies.

This proposed system offers a cost-effective, end-to-end solution that replaces the need for multiple disjointed tools. By automating the analysis process and providing rich, interpretable data, it empowers athletes and coaching staff with actionable insights to enhance performance and strategy.

Chapter 6

Experimental Setup

This study presents a multi-model approach for automated badminton match analysis, leveraging deep learning techniques for non-reply detection, player detection, shuttlecock tracking, and pose estimation. The system consists of four primary components:

1. Non-Reply Detection: A CNN-based model processes frame timestamps to classify rally terminations.
2. Player Detection: YOLO (You Only Look Once) detects players in real-time.
3. Shuttlecock Detection and Tracking: TrackNet is utilized for tracking shuttlecock movement across frames.
4. Pose Estimation: A pose classification model analyzes player movement and actions.

Each of these components is trained and tested on specific datasets with optimized hyperparameters to ensure high accuracy, robustness, and real-time efficiency.

6.1 Dataset Collection and Preprocessing

6.1.1. Non-Replay Frame Detection Dataset

- Input: Frame timestamps.
- Collected from annotated match footage with rally labels.
- Preprocessing: Normalized timestamp values converted into sequential arrays for CNN input.
- CNN model architecture is shown in Table 1.

Layer (Type)	Output Shape	Param #
conv2d (Conv2D)	(None, 106, 190, 30)	300
max_pooling2d (MaxPooling2D)	(None, 53, 95, 30)	0
conv2d_1 (Conv2D)	(None, 51, 93, 30)	8,130
max_pooling2d_1 (MaxPooling2D)	(None, 25, 46, 30)	0
flatten (Flatten)	(None, 34500)	0
dense (Dense)	(None, 128)	4,416,128
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Table 6.1 Architecture of the CNN Model

6.1.2. Player Detection Dataset

- Model Used: YOLOv8.
- Annotations: Bounding boxes for player positions.
- Preprocessing:
    - Resizing images to 640×640 px for YOLO compatibility.
    - Data augmentation (random crop, rotation, brightness adjustments).
    - Label conversion to YOLO format (object class, x-center, y-center, width, height).

Fig 6.1 Object Detection for Players by YOLOv8

6.1.3. Shuttlecock Detection Dataset

- Model Used: TrackNetV3
- Dataset: Sequential badminton video frames annotated with shuttlecock positions.
- Preprocessing:
  - Frame extraction at 30 FPS.
  - Shuttlecock positions annotated as (x, y) coordinates.
  - Visibility Readings also given using 1 and 0 for each frame.

Fig 6.2 TrackNetV3 Shuttlecock Detections

6.1.4. Pose Estimation Dataset

- Model Used: OpenPose-based network.
- Dataset: Badminton Pose Classification dataset from Roboflow.
- Preprocessing:
  - Scaling and normalizing keypoints.
  - Synthetic data augmentation (pose variations, mirroring, jittering).

6.2 Deep Learning Models Used

6.2.1. Non-Reply Detection (CNN)

- Architecture: Total: 8 Layers (Excluding Input Layer). If we count only trainable layers, then Dropout is not included, making it 7 trainable layers.
- Input: Processed timestamps.
- Output: Rally termination classification.
- Activation Functions: ReLU and Softmax.
- Loss Function: Categorical Cross-Entropy.

6.2.2. Player Detection (YOLOv8)

- Backbone: CSPDarkNet53.
- Input Size: 640×640.
- Loss Function: CIoU Loss.
- Training:
    - Batch Size: 16
    - Learning Rate: 0.001
    - Optimizer: AdamW

6.2.3. Shuttlecock Tracking (TrackNetV3)

- Input: 3 consecutive frames.
- Output: Heatmap indicating shuttlecock position.
- Loss Function: Mean Squared Error (MSE).
- Optimizer: Adam.
- Learning Rate: 0.0001.

6.2.4. Pose Estimation (OpenPose-Based Model)

- Backbone: ResNet50.
- Input: Preprocessed player frames.
- Output: 17 body keypoints with action classification.
- Loss Function: Mean Squared Error + Categorical Cross-Entropy.
- Optimizer: RMSprop.
- Learning Rate: 0.0001.

6.3 Training Methodology and Hyperparameters

- Dataset Split: 80% Training | 10% Validation | 10% Testing.
- Augmentation Techniques: Rotation, flipping, color jittering.
- Hardware Used:
    - GPU: NVIDIA RTX 3090 (24GB VRAM).
    - CPU: Intel i9-12900K.

- RAM: 32GB DDR5.
- Storage: 1TB NVMe SSD.

- Software & Libraries:
    - TensorFlow 2.x / PyTorch 2.0.
    - OpenCV for video processing.
    - Ultralytics YOLOv8 framework.
    - CUDA 11.8 for GPU acceleration.
    - Roboflow for inference models.

Component	Specification
GPU	NVIDIA RTX 3090 (24GB VRAM)
CPU	Intel i9-12900K
RAM	32GB DDR5
Storage	1TB NVMe SSD
Deep Learning Framework	TensorFlow 2.x / PyTorch 2.0
Computer Vision Library	OpenCV
Optimization Library	CUDA 11.8

Table 6.2 Hardware and Software Specifications

6.4 Performance Metrics and Evaluation Strategy

6.4.1. CNN for Non-Reply Detection

- Metrics Used:
    - Accuracy: Measures correct rally terminations.
    - Loss Reduction: Analyzes model convergence.
- Results:
    - Training loss decreases steadily.
    - Validation loss remains stable (no overfitting).

6.4.2. YOLO for Player Detection

- Metrics Used:
    - mAP (mean Average Precision) at IoU 0.5.
    - Precision and Recall.
- Results:
    - mAP: 0.91.

6.4.3. TrackNet for Shuttlecock Tracking

- Metrics Used:
    - IoU (Intersection over Union) for shuttlecock bounding box accuracy.
    - Frame-wise tracking accuracy.
- Results:
    - IoU: 0.87.
    - Frame-wise tracking precision: 92.3%.

6.4.4. Pose Estimation Model

- Metrics Used:
    - Keypoint Accuracy: Measures correct pose detection.
    - Classification Accuracy: Evaluates predicted action labels.
- Results:
    - mAP: 89.6%.
    - Precision: 82.0%.
    - Recall: 83.8%

Models	**Accuracy**	**mAP50**	**Precision**	**Recall**
CNN	93.0%	92.5%	95.3%	91.4%
YOLOv8	79.0%	82.3%	80.3%	77.6%
TrackNetV3	97.51%	-	97.79%	99.33%
Roboflow Pose Predictor	82.9%	89.6%	82.0%	83.8%

Table 6.3 Architecture of the CNN Model

TrackNetV3 achieved the highest accuracy (97.51%) and recall (99.33%), making it the most effective model overall as it is also the most crucial module of the entire system. CNN performed well with 93.0% accuracy and high precision (95.3%) ideal for Play and Non-Play frame classification. YOLOv8 had lower accuracy (79.0%) but decent precision (80.3%). Roboflow Pose Predictor showed moderate performance with 82.9% accuracy and 89.6% mAP50, making it a good choice for the proposed system, although there is room for improvement.

Chapter 7

Results and Discussion

7.1 System Performance and Processing Time

The developed system demonstrates a processing time of approximately 2-3 minutes to analyze a full badminton match segment. This includes non-reply detection, player detection, shuttlecock tracking, and pose estimation. The time efficiency is primarily influenced by hardware specifications and the deep learning models' complexity, with GPU acceleration playing a crucial role in reducing inference time.

Fig 7.1 Output Result Frame Sample

7.2 CNN-Based Non-Reply Detection Results

The Convolutional Neural Network (CNN)-based non-reply detection model exhibits consistent convergence over training epochs, as indicated by the loss curves in the image. The box loss, classification loss, and distribution focal loss (DFL loss) show a downward trend, confirming effective learning. Meanwhile, validation losses follow a similar trajectory, ensuring minimal overfitting. The precision and recall metrics improve steadily across epochs, with final values reaching above 0.8 and 0.5, respectively. The mean Average Precision (mAP50) metric stabilizes around 0.5, indicating strong detection capability.

7.3 YOLO-Based Player Detection and Pose Estimation

The YOLO model efficiently detects players in real-time, with detection precision improving progressively during training. Pose estimation, handled using OpenPose or a similar model, achieves accurate skeletal representations of players, assisting in action classification and shot prediction.

7.4 Shuttlecock Tracking Using TrackNet

Shuttlecock tracking, powered by TrackNet, performs reliably under standard conditions but may exhibit minor inaccuracies in fast-moving sequences due to motion blur. Future enhancements could involve higher frame-rate data augmentation to improve model robustness.

7.5 Discussion on System Accuracy and Optimization

- Real-time Processing Feasibility: While the system currently takes 2-3 minutes per match segment, further optimization via model quantization, TensorRT acceleration, or pruning techniques could enhance efficiency.
- Impact of Hyperparameters: The use of optimized learning rates, dropout layers, and batch normalization ensures stable training and prevents overfitting.
- Generalization Ability: The models generalize well across different match scenarios, but potential biases due to dataset limitations could be addressed with diverse training data.

Chapter 8

Conclusion and Future Work

The proposed system integrates deep learning-based non-reply detection, player tracking, shuttlecock tracking, and pose estimation to analyze badminton matches effectively. The CNN-based non-reply detection model achieves promising results, with progressive loss reduction and improved precision-recall performance. The YOLO-based player detection and TrackNet-powered shuttlecock tracking ensure robust motion analysis, while pose estimation enhances player movement understanding.

Despite achieving high accuracy and stable performance, the system currently requires 2-3 minutes per match segment, making real-time processing challenging. The results indicate that deep learning techniques are highly effective for badminton match analysis, but further optimization is necessary to enhance processing speed and generalizability across diverse match environments.

To improve efficiency and accuracy, the following enhancements are planned:

1. **Real-Time Optimization:**
   - Implement model quantization (e.g., TensorRT, ONNX) to speed up inference.
   - Utilize pruning and knowledge distillation to reduce model size while retaining accuracy.
   - Explore hardware acceleration (e.g., TPUs, optimized GPUs) for faster computations.
2. **Enhanced Data Collection and Augmentation:**
   - Increase dataset diversity by including matches from different angles, lighting conditions, and court types.
   - Implement data augmentation techniques to improve robustness against motion blur and occlusions.
3. **Pose-Based Action Recognition:**
   - Use Graph Neural Networks (GNNs) to model player poses and classify actions more accurately.
   - Train on larger-scale action recognition datasets for better generalization.
4. **Explainability and Visualization:**
   - Develop real-time overlays for match analysis, showing detected events, player positions, and shuttlecock trajectory.
   - Implement a user-friendly interface for match playback with automated insights and statistics.

With these enhancements, the system can be transformed into a real-time AI-powered badminton analytics tool useful for coaches, players, and sports analysts, offering insights for performance improvement, strategy analysis, and game intelligence.

References

[1] Y.-H. Hsu, C.-C. Yu, and H.-Y. Cheng, “Enhancing Badminton Game Analysis: An Approach to Shot Refinement via a Fusion of Shuttlecock Tracking and Hit Detection from Monocular Camera,” Sensors, vol. 24, no. 13, p. 4372, 2024. doi: 10.3390/s24134372.

[2] Y. Nokihara, R. Hachiuma, R. Hori, and H. Saito, “Future Prediction of Shuttlecock Trajectory in Badminton Using Player’s Information,” Journal of Imaging, vol. 9, no. 5, p. 99, 2023. doi: 10.3390/jimaging9050099.

[3] A. Kumar and P. V. S. S. R. C. Mouli, “Vision-Based Automated Badminton Action Recognition Using the New Local Convolutional Neural Network Extractor,” International Journal of Computer Applications, vol. 175, no. 15, pp. 1–6, 2020. doi: 10.5120/ijca2020920159.

[4] Y.-C. Chen and C.-H. Men, “Automatic Shuttlecock Fall Detection System in or out of a Court in Badminton Games—Challenges, Problems, and Solutions from a Practical Point of View,” Sensors, vol. 22, no. 21, p. 8098, 2022. doi: 10.3390/s22218098.

[5] W.-C. Wang and L.-C. Fu, “Real-Time Badminton Player Tracking Using Computer Vision,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 10, pp. 3541–3554, 2020. doi: 10.1109/TCSVT.2020.2976673.

[6] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng, “TrackNet: A Deep Learning Network for Tracking High-Speed and Tiny Objects in Sports Applications,” arXiv preprint arXiv:1907.03698, 2019.

[7] P. Liu and J.-H. Wang, “MonoTrack: Shuttle Trajectory Reconstruction from Monocular Badminton Video,” arXiv preprint arXiv:2204.01899, 2022. [8] Y.-H. Chien and F. Yu, “Automated Hit-Frame Detection for Badminton Match Analysis,” arXiv preprint arXiv:2307.16000, 2023.

[8] Y.-H. Chien and F. Yu, “Automated Hit-Frame Detection for Badminton Match Analysis,” arXiv preprint arXiv:2307.16000, 2023.

[9] Q. Li, T.-C. Chiu, H.-W. Huang, M.-T. Sun, and W.-S. Ku, “VideoBadminton: A Video Dataset for Badminton Action Recognition,” *arXiv preprint arXiv:2403.12385*, 2024.

[10] P.-Y. Chou, Y.-C. Lo, B.-Z. Xie, C.-H. Lin, and Y.-Y. Kao, “An All Deep System for Badminton Game Analysis,” *arXiv preprint arXiv:2308.12645*, 2023.

[11] C.-T. Wei and S.-K. Weng, “A court line extraction algorithm for badminton tournament videos with horizontal line projection learning,” *IET Image Processing*, vol. 17, no. 10, pp. 2907–2924, 2023.

[12] J. Liu and B. Liang, “An action recognition technology for badminton players using deep learning,” *Mobile Information Systems*, vol. 2022, no. 1, p. 3413584, 2022.

[13] C. Cai, “Badminton video analysis based on player tracking and pose trajectory estimation,” in *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 471–474, IEEE, 2021.

[14] K.-C. Lin, C.-W. Ko, H.-C. Hung, and N.-S. Chen, “The effect of real-time pose recognition on badminton learning performance,” *Interactive Learning Environments*, vol. 31, no. 8, pp. 4772–4786, 2023.

[15] D. Y. W. Tan, H. Y. Ting, and S. B. Y. Lau, “A review on badminton motion analysis,” in *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, pp. 1–4, IEEE, 2016.

[16] M.-Á. Gómez, F. Rivas, A. S. Leicht, and J. M. Buldú, “Using network science to unveil badminton performance patterns,” *Chaos, Solitons & Fractals*, vol. 135, p. 109834, 2020.

[17] J. Zhang and W. Xu, “Deep Learning for Sports Analytics: A Comprehensive Survey,” *IEEE Transactions on Multimedia*, 2021.

[18] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[19] C. Wang, A. Bochkovskiy, and H. M. Liao, “YOLOv8: Towards Real-Time Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[20] Y. Huang and H. Fu, “TrackNet: A Deep Learning Model for Shuttlecock Tracking in Badminton Matches,” *IEEE Transactions on Image Processing*, 2020.

[21] J. Lin and S. Lee, “Pose Estimation in Sports Using Deep Learning: A Review,” *Computer Vision and Image Understanding*, 2019.

Appendix

Publication

Appendix – List of Publications or certificates

Acknowledgements

We would like to express our sincere gratitude to all those who supported and guided us throughout the course of this project.

First and foremost, we want to extend our heartfelt thanks to our project supervisor, Dr. Jaya Gupta, for their invaluable guidance, constant encouragement, and insightful feedback at every stage of this work. Their expertise and mentorship have been crucial in shaping the direction and outcome of this project.

We are also thankful to A.P. Shah Institute of Technology, the Department of Computer Science Engineering, Artificial Intelligence and Machine Learning, for providing the resources and environment necessary for carrying out this research.

This project would not have been possible without the collective contributions and encouragement from everyone involved in this team.