Volleyball Performance Statistics Based on Video Analysis and Deep Learning

Bruno Filipe Baptista Ferreira

Supervisors: José Manuel Torres and Christophe Soares

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

A dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Science in Mobile Computing

2026

Abstract

This dissertation presents the development and evaluation of a semi-automatic video-based system for extracting volleyball performance statistics using computer vision and deep learning techniques. Performance analysis plays a critical role in modern volleyball, supporting coaches and analysts in understanding match dynamics, technical execution, and tactical behaviour. However, the collection of such statistics is often performed manually or semi-manually, making the process time-consuming, difficult to scale, and prone to inconsistency.

To address these limitations, this work proposes a modular analysis pipeline centred on ball detection, tracking, scoreboard reading, court calibration, rally management, and event interpretation. The system employs a YOLO-based model for volleyball detection, combined with temporal tracking strategies that exploit spatial continuity, motion constraints, foreground validation, and outlier rejection. Manual initialisation is still required for court calibration and scoreboard region selection, but the subsequent analysis stage operates automatically over the selected video segment.

The work integrates these components into a unified prototype for volleyball match analysis. The system produces structured outputs describing rallies and event categories such as attacks, blocks, balls out, balls on the net, aces, and errors. The implemented pipeline was evaluated on two recorded volleyball sets from different venues. The system achieved a manual-review agreement rate of 81.82% on the first evaluated set and 72.50% on the second, corresponding to an overall agreement rate of 77.38% across 84 reviewed points.

The evaluation shows that the proposed approach can extract useful volleyball event information from fixed-camera match footage, while also highlighting important limitations. The main sources of disagreement were related to fast rallies, trajectory ambiguity near the net and court boundaries, visual interference from additional balls, and venue-specific background complexity. The prototype therefore offers a concrete starting point for future work on semi-automatic volleyball performance analysis, particularly through improved ball tracking, more robust scoreboard recognition, and broader evaluation across additional venues.

Resumo

Esta dissertação apresenta o desenvolvimento e a avaliação de um sistema semiautomático baseado em vídeo para a extração de estatísticas de desempenho em voleibol, utilizando técnicas de visão computacional e aprendizagem profunda. A análise de desempenho desempenha um papel importante no voleibol moderno, apoiando treinadores e analistas na compreensão da dinâmica do jogo, da execução técnica e do comportamento tático. No entanto, a recolha deste tipo de estatísticas é frequentemente realizada de forma manual ou semimanual, tornando o processo moroso, difícil de escalar e sujeito a inconsistências.

Para responder a estas limitações, este trabalho propõe uma pipeline modular de análise centrada na deteção da bola, tracking, leitura do marcador, calibração do campo, gestão de rallies e interpretação de eventos. O sistema utiliza um modelo baseado em YOLO para a deteção da bola de voleibol, combinado com estratégias de tracking temporal que exploram a continuidade espacial, restrições de movimento, validação por foreground e rejeição de outliers. A inicialização manual continua a ser necessária para a calibração do campo e para a seleção da região de interesse do marcador, mas a fase posterior de análise decorre automaticamente sobre o segmento de vídeo selecionado.

A integração destes componentes resulta num protótipo unificado para análise de jogos de voleibol. O sistema produz saídas estruturadas que descrevem rallies e categorias de eventos, tais como ataques, blocos, bolas fora, bolas na rede, ases e erros. A pipeline implementada foi avaliada em dois sets de voleibol gravados em pavilhões diferentes. O sistema obteve uma taxa de concordância com a revisão manual de 81.82% no primeiro set avaliado e 72.50% no segundo, correspondendo a uma taxa global de concordância de 77.38% em 84 pontos revistos.

A avaliação mostra que a abordagem proposta consegue extrair informação útil sobre eventos de voleibol a partir de filmagens de jogo com câmara fixa, ao mesmo tempo que evidencia limitações importantes. As principais fontes de desacordo estiveram relacionadas com rallies rápidos, ambiguidades de trajetória junto à rede e aos limites do campo, interferência visual causada por bolas adicionais e diferenças de complexidade visual do fundo entre

pavilhões. O protótipo constitui assim um ponto de partida concreto para trabalho futuro em análise semiautomática de desempenho em voleibol, particularmente através da melhoria do tracking da bola, de uma leitura mais robusta do marcador e de uma avaliação mais ampla em pavilhões adicionais.

I would like to dedicate this dissertation to my parents and family, for their endless love and encouragement

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. José Torres and Prof. Christophe Soares, for their guidance, support, and valuable insights throughout the development of this dissertation. Their feedback and direction were essential in shaping this work.

I would also like to thank the professors of the Mobile Computing course at Universidade Fernando Pessoa for the knowledge and skills they provided, which were fundamental to the completion of this project.

My sincere thanks go to my family, friends, and colleagues for their continuous support, encouragement, and motivation during this journey.

Contents

List of Figures

List of Tables

Acronyms

AI *Artificial Intelligence*

CNN *Convolutional Neural Network*

COCO *Common Objects in Context*

CSV *Comma-Separated Values*

CV *Computer Vision*

DNN *Deep Neural Network*

FIVB *Fédération Internationale de Volleyball*

FPS *Frames Per Second*

GPU *Graphics Processing Unit*

JSON *JavaScript Object Notation*

MOG2 *Mixture of Gaussians 2*

OCR *Optical Character Recognition*

RANSAC *Random Sample Consensus*

R-CNN *Region-based Convolutional Neural Network*

Re-ID *Re-identification*

ROI *Region of Interest*

VIS *Volleyball Information System*

YOLO *You Only Look Once*

Chapter 1

Introduction

The introduction frames the need for a semi-automatic volleyball video analysis system and explains why the selected application scenario requires a combination of object detection, tracking, scoreboard reading, and event classification.

1.1 Motivation

The systematic analysis of athletic performance has become an indispensable component of modern competitive sport. In team sports, coaches and analysts rely on detailed statistical records to evaluate individual and collective performance, identify tactical patterns, prepare for upcoming opponents, and monitor player development over time. In volleyball specifically, the set of statistics of interest is well established and includes indicators such as serve efficiency, reception quality, attack success rate, block effectiveness, and rally duration (Silva et al., 2016); (Fédération Internationale de Volleyball, 2026); (.Işgüzar et al., 2026). These metrics inform decisions at every level of the game, from real-time tactical adjustments during a match to long-term programme planning between seasons.

Historically, the collection of these statistics has depended on manual notational analysis: a trained analyst observes match footage and records events by hand or using dedicated software such as DataVolley. While this approach benefits from expert contextual judgement, it is inherently time-consuming, difficult to standardise across analysts, and practically limited by human attention over extended video sequences (Cheng et al., 2022); (Fujii, 2025). In a competitive environment where multiple matches may be played each week, manual annotation creates a significant operational bottleneck and may delay the delivery of feedback to players and coaching staff.

Advances in computer vision and deep learning have opened a credible path towards automating this process (Fujii, 2025). The emergence of fast and accurate object detectors, robust temporal tracking algorithms, and increasingly powerful hardware has made it feasible to extract structured information directly from raw match video without requiring continuous human input. A semi-automatic analysis pipeline, requiring only a short manual setup before processing the selected segment, carries practical value for clubs, federations, and research institutions that lack the resources to employ dedicated analysis staff.

This dissertation is motivated by the specific challenge of realising such a pipeline for volleyball, applied to real match footage recorded in a Portuguese indoor venue. The target footage introduces a set of technically demanding conditions: a small and fast-moving ball that frequently occupies fewer than 10×10 pixels in the frame, a complex background including a visually prominent mural that generates persistent false-positive detections, variable illumination, and an on-screen scoreboard whose automatic reading is essential for reliable rally boundary detection. Addressing these challenges requires integrating solutions from multiple sub-fields of computer vision — object detection, temporal tracking, optical character recognition, and geometric calibration — into a coherent and practically deployable system.

Figure 1.1 summarises the motivation for the proposed system, illustrating the transition from manual annotation to a structured video-analysis pipeline.

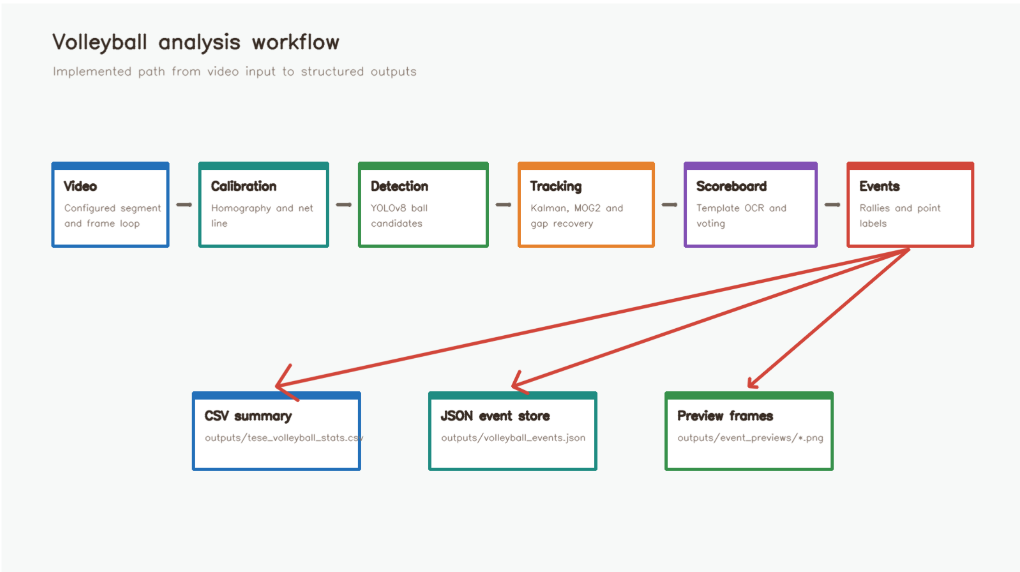


Figure 1.1: Overview of the proposed volleyball analysis workflow, showing how recorded match footage is transformed into structured rally and event information through detection, tracking, scoreboard reading, and event classification.

1.2 Problem Statement

The central problem addressed in this dissertation is the semi-automatic extraction of volleyball match statistics from a single fixed-camera video recording of a real competitive match, with manual intervention limited to an initial one-time calibration step.

More precisely, the system must be capable of:

1. **Detecting the volleyball** reliably across the full match duration, including frames in which the ball is partially occluded, subject to motion blur, or visually similar to background elements.
2. **Tracking the ball's trajectory** across consecutive frames, maintaining temporal continuity through short gaps caused by occlusion or missed detections, and producing a spatially consistent trajectory that faithfully represents the ball's physical path.
3. **Reading the match scoreboard** automatically from the video signal, detecting score changes that mark the end of each rally, and doing so robustly despite the noise inherent in per-frame optical character recognition.
4. **Classifying rally-level events**, including spikes, blocks, aces, and unforced errors, by analysing the ball's trajectory relative to the court geometry and the position of the net.
5. **Producing structured statistical output** in standard formats (CSV and JSON) that can be inspected, visualised, and integrated into broader analysis workflows.

The system is designed to operate on pre-recorded footage in a near-offline mode, where real-time performance is desirable but not strictly required. The primary constraint is accuracy: a pipeline that produces statistics with systematic biases or high error rates would not deliver practical value, regardless of its speed.

1.3 Objectives

The specific objectives of this dissertation are:

1. To design and implement a modular computer vision pipeline for semi-automatic volleyball match analysis, encompassing ball detection, temporal tracking, court geometry calibration, scoreboard reading, and event classification.
2. To construct a custom annotated dataset from the target match footage and to implement a ball detection module suitable for the visual conditions of volleyball matches, including mechanisms to suppress venue-specific false positives.
3. To develop a robust scoreboard reading module that integrates optical character recognition with domain-specific volleyball rules, enabling reliable rally boundary detection under noisy visual conditions.
4. To implement and evaluate event classification logic for the most technically significant volleyball actions — spike, block, and ace — using trajectory analysis relative to calibrated court geometry.

5. To evaluate the system on real match footage using manual review of rally-level event classifications, reporting agreement rates and analysing the main sources of disagreement.

1.4 Contributions

The main technical contributions of this dissertation are summarised in Table 1.1.

Table 1.1: Main technical contributions.

Contribution	**Description**
End-to-end analysis pipeline	Integrates ball detection, Kalman filter-based tracking, homographic court calibration, template-based scoreboard OCR, and event classification into a unified modular system for uncontrolled single-camera footage.
Domain-specific detector	Uses manually annotated frames from a Portuguese volleyball league match, including venue-specific background examples, to reduce false positives caused by the sports hall mural and other distractors.
Rule-aware scoreboard OCR	Combines template-based digit reading with volleyball scoring constraints, temporal voting, and digit locking to stabilise noisy score readings and infer rally boundaries.
Trajectory-based event inference	Uses calibrated ball trajectories and tracking gaps near the net to infer rally-ending events such as attacks, blocks, aces, balls out, and balls on the net.
Documented custom dataset	Provides an annotated YOLO-format volleyball ball dataset from an indoor venue to support reproducibility and future extension.

1.5 Scope and Limitations

The system developed in this dissertation is scoped to fixed-camera indoor volleyball recordings from real club-level venues. Most development and parameter tuning were carried out using footage from *Pav. Arq. Jerónimo Reis Espinho*, while some screenshots and walkthrough examples were generated from the active footage at the time of documentation, recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. The calibration procedure, scoreboard template library, and detection model remain venue-dependent. While the architectural design is intended to generalise to other venues and camera configurations with appropriate recalibration and retraining, such generalisation remains limited and is considered future work.

Player detection and tracking are outside the scope of this dissertation. The pipeline does not identify individual players, estimate player poses, or attribute events to specific athletes. All event classification is performed solely from ball trajectory information relative to court geometry.

The system is not designed for real-time deployment in the sense of on-court live feedback. Processing is performed offline on recorded video, and the pipeline is optimised for accuracy rather than minimum latency, though practical throughput on consumer-grade GPU hardware was a secondary consideration in architectural decisions.

1.6 Dissertation Structure

The remainder of this dissertation is organised as follows.

Chapter 2 reviews the relevant literature on performance analysis in volleyball, computer vision for sports analytics, deep learning-based object detection, ball tracking, scoreboard reading, and publicly available datasets for sports vision tasks. It then positions this work relative to prior contributions and identifies the gap addressed by this dissertation.

Chapter 3 specifies the application scenario, intended use case, target footage, functional requirements, pipeline architecture, and design decisions that guide the proposed solution. It also discusses court calibration, scoreboard region selection, and the data flow between the main system components.

Chapter 4 details the implementation of the proposed system, including the development environment, configuration files, court calibration procedure, custom dataset preparation, YOLO-based ball detector, tracking and validation logic, block detection module, scoreboard reader, rally management, event classification rules, and output generation mechanisms.

Chapter 5 evaluates the implemented system on two recorded volleyball sets from different venues. The analysis is based on manual review of the system's event classifications and reports agreement rates for each match and overall, together with the main sources of disagreement: fast rallies, ambiguity near the net and court boundaries, visual interference from additional balls, and venue-specific differences.

Chapter 6 summarises the main contributions, reflects on the limitations of the implemented system and its evaluation, and outlines possible directions for future work, including improved ball tracking, more robust scoreboard recognition, broader multi-venue evaluation, and extensions towards player tracking or real-time deployment.

Overall, this introductory Chapter establishes the practical motivation for the work, defines the research problem, presents the research objectives, and sets out the structure followed by the remaining Chapters.

Chapter 2

Background and Related Work

This Chapter reviews the literature and existing work relevant to the semi-automatic analysis of volleyball matches from video. The review is organised around five interconnected areas that underpin the system developed in this dissertation: (1) performance analysis in volleyball and sports analytics; (2) object detection with deep learning for sports video; (3) ball detection, temporal continuity, and tracking; (4) scoreboard reading in sports broadcasts; and (5) publicly available datasets for volleyball and sports vision. Each section identifies key contributions in the relevant body of work, discusses the technical challenges that motivate design choices, and justifies the relevance of each area to the problem at hand. The Chapter concludes with a summary of the literature landscape and a positioning of this dissertation relative to existing work.

2.1 Performance Analysis in Volleyball

Performance analysis in team sports has undergone substantial change over the past two decades, driven by the increasing availability of video recordings, the emergence of specialised annotation software, and, more recently, advances in computer vision and machine learning. In volleyball specifically, statistical indicators such as serve efficiency, reception quality, attack success rate, block effectiveness, and rally duration are routinely used by coaching staff to inform tactical decisions, evaluate individual player contributions, and prepare for upcoming opponents.

Historically, these statistics were gathered through manual notational analysis: a trained observer reviews match footage and records events by hand or using dedicated software. This approach benefits from expert contextual understanding but presents well-documented practical limitations. It is time-consuming, difficult to replicate consistently across different analysts, and fundamentally constrained by human attention over extended video sequences (Cheng et al., 2022); (Fujii, 2025). In competitive environments where multiple matches are played each week, manual annotation creates a significant operational bottleneck and may delay feedback to players and coaching staff.

Semi-automatic platforms such as DataVolley have become widely used in professional volleyball, allowing analysts to tag events in near-real-time by pressing predefined shortcuts while observing footage (Cheng et al., 2022). These systems reduce some of the annotation burden but still require a trained human operator who actively identifies and labels each event. Information is not extracted directly from the raw video signal; the quality of the resulting statistics therefore remains dependent on individual operator expertise.

Research efforts aimed at fully automatic video-based volleyball analysis have addressed different aspects of the problem with varying degrees of success. Early work by Ibrahim et al. (2016) proposed methods for activity detection in indoor sports, including volleyball, by analysing player silhouettes and motion patterns extracted from fixed-camera recordings. Although this work demonstrated the feasibility of modelling individual and collective volleyball activities from video, it was primarily designed for group activity recognition rather than direct ball trajectory extraction or full match statistics generation.

The recognition of collective actions has also received attention. Ibrahim et al. (2018) introduced a hierarchical relational model for group activity recognition in volleyball that processes both individual player movements and their spatial relationships to infer higher-level tactical events. This work is particularly relevant because it acknowledges the relational nature of volleyball actions: a block, for instance, cannot be identified from a single player in isolation but requires understanding the trajectory of the ball relative to multiple players on both sides of the net.

Ball trajectory analysis has emerged as a central task in volleyball video understanding, since the ball mediates all significant game events. Huang (2023) presented a method combining volleyball trajectory estimation with embedded graph convolution, demonstrating that spatial relationships between consecutive ball positions can support trajectory analysis and action-related interpretation.

Taken together, the existing literature establishes that automatic volleyball analysis from video is technically feasible but challenging. The difficulties identified consistently across works include the small apparent size of the ball, high-speed motion causing motion blur, frequent occlusions by players and the net, and variability of visual conditions across venues and camera setups.

2.2 Object Detection with Deep Learning for Sports Video

Object detection is the computer vision task of simultaneously localising and classifying one or more objects within an image, typically by predicting bounding boxes and associated class labels. For sports video analysis, detection must contend with fast motion, partial occlusion, cluttered backgrounds, and objects that may be very small in relation to the frame. The evolution of deep learning-based detection has progressively improved robustness to these conditions.

2.2.1 From Classical Methods to Deep Learning

Prior to the widespread adoption of deep neural networks, object detection relied on hand-crafted feature descriptors such as Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) combined with sliding window classifiers or deformable part models. While effective in controlled settings, these approaches proved fragile when applied to objects with complex appearance variation, high background clutter, or significant motion blur — conditions that are systematically present in indoor volleyball footage.

The introduction of Region-based Convolutional Neural Networks (R-CNN) by Girshick et al. (2014) marked a significant turning point, establishing a two-stage paradigm in which candidate regions are first proposed and then individually classified by a convolutional network. Subsequent variants, including Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015), improved computational efficiency substantially while maintaining competitive accuracy. Two-stage detectors remain relevant for precision-critical applications, and Faster R-CNN has been made accessible through frameworks such as Detectron2. However, their inference overhead limits applicability in scenarios requiring high frame-rate processing.

2.2.2 Single-Stage Detectors and the YOLO Family

The You Only Look Once (YOLO) architecture, first proposed by Redmon et al. (2016), reformulates object detection as a single regression problem: the input image is divided into a grid, and each cell directly predicts bounding boxes, class probabilities, and confidence scores in a single forward pass. This eliminates the region proposal stage entirely, substantially reducing inference latency and making the family of architectures suitable for real-time or near-real-time applications.

YOLOv3 (Redmon and Farhadi, 2018) introduced multi-scale prediction using feature maps at three different resolutions, significantly improving the detection of small objects compared to earlier versions. This is particularly relevant for ball detection in volleyball, where the ball may occupy fewer than 10×10 pixels in a standard broadcast frame. The multi-scale mechanism allows the detector to allocate dedicated capacity to small-object localisation without sacrificing accuracy on larger objects.

Successive iterations of the YOLO lineage have continued to improve detection accuracy and flexibility. YOLOv8, released by Ultralytics (Jocher et al., 2024), introduced an anchor-free detection head, which simplifies training and improves generalisation to objects with variable aspect ratios. It is available in multiple size variants, allowing practitioners to trade detection accuracy for inference speed while maintaining the practical advantages of the YOLO family for high-throughput sports video analysis.

2.2.3 Challenges in Small Object Detection

Small object detection remains an active research challenge. Objects occupying fewer than 32×32 pixels in the input image provide limited gradient signal during training and are more susceptible to suppression by non-maximum suppression when in proximity to larger, higher-confidence detections. Several strategies have been proposed to address this:

- **High-resolution inference**: processing images at larger input sizes preserves spatial detail for small objects, at a computational cost that is acceptable in offline analysis pipelines.
- **Feature Pyramid Networks (FPN)** (Lin et al., 2017): multi-scale feature aggregation enables the detector to combine high-resolution spatial information from early layers with high-level semantic information from deeper layers, improving recall for small objects.
- **Data augmentation**: techniques such as mosaic augmentation, random scaling, and copy-paste help the model generalise to object sizes and positions not well-represented in the training set.
- **Hard negative mining**: including frames in which no target object is present as background images during training explicitly teaches the model to reject common false-positive regions, reducing spurious detections from visually confusable scene elements (Shrivastava et al., 2016).

The combination of high-resolution inference and hard negative mining is particularly relevant for volleyball ball detection, where the risk of confusing the ball with other circular or spherical elements in indoor venues is non-trivial. For this reason, the pipeline emphasises domain-specific fine-tuning and targeted negative examples rather than relying only on generic pretrained detector behaviour.

2.3 Ball Detection, Temporal Continuity, and Tracking

Per-frame object detection provides a localisation estimate for each individual frame but does not by itself produce a temporally coherent trajectory. In practice, detectors applied to volleyball footage will produce both missing detections — due to motion blur, occlusion, or the ball passing near a visually similar region — and spurious false positives due to background distractors. Tracking algorithms address both problems by exploiting temporal continuity: the assumption that the ball's position at frame $t+1$ can be predicted from its position and velocity at earlier frames.

2.3.1 The Tracking-by-Detection Paradigm

The dominant paradigm in modern object tracking is tracking-by-detection: a per-frame detector is applied independently to each frame, and the resulting detections are linked across frames by a data association algorithm to form trajectories. Association is typically performed using distance metrics — Euclidean distance between predicted and observed positions, or intersection over union (IoU) between bounding boxes — combined with optimal assignment algorithms such as the Hungarian method (Kuhn, 1955). This paradigm cleanly separates the detection problem from the association problem, allowing each to be improved independently.

Figure 2.1 illustrates this tracking-by-detection principle by showing how per-frame detections are linked over time and how short missing intervals can be bridged by prediction.

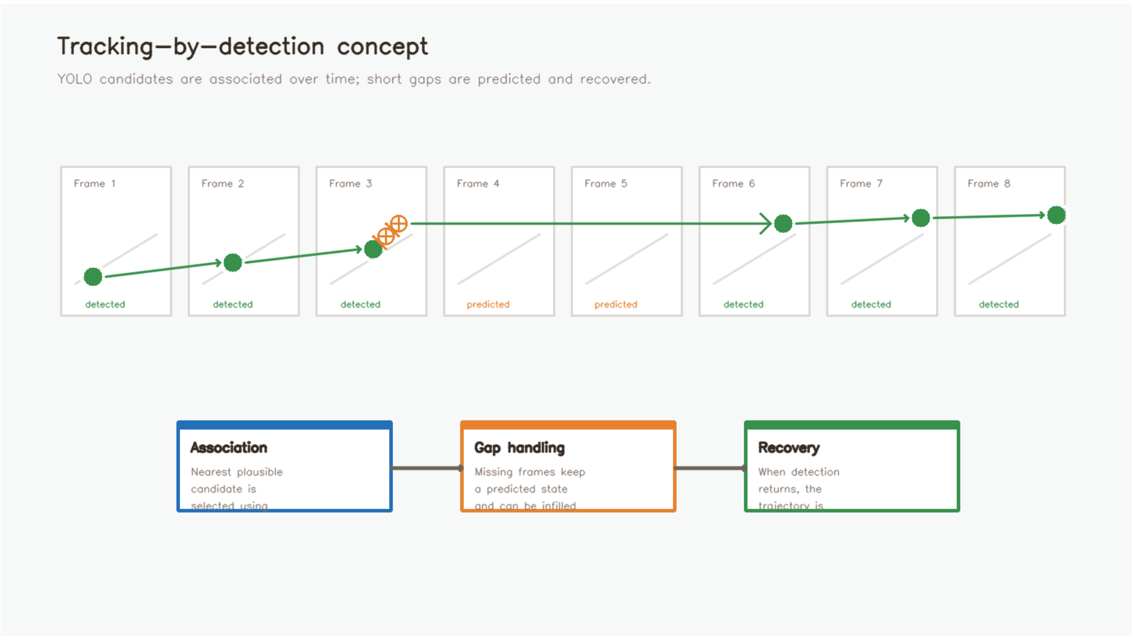


Figure 2.1: Tracking-by-detection concept.

In the sports domain, the tracking-by-detection paradigm has been applied to both player tracking and ball tracking, but ball tracking presents distinctive challenges: the ball moves faster than players, undergoes more dramatic changes in apparent size due to perspective, is subject to severe motion blur at high speeds, and may be invisible for extended periods during occlusion. TrackNet (Huang et al., 2019), developed for tennis ball tracking, addressed some of these challenges by framing tracking as a heatmap prediction problem over a short temporal window of consecutive frames, effectively encoding temporal context within the network architecture itself. This approach has influenced subsequent work on ball tracking in other sports.

For multi-object tracking at the level of players, recent benchmarks such as SportsMOT (Cui et al., 2023), introduced at ICCV 2023, have provided standardised evaluation conditions for tracking in team sports including volleyball. SportsMOT contains sequences with dense bounding box annotations and persistent identity labels in the MOT Challenge format, enabling rigorous comparison of tracking approaches under realistic sports conditions including fast motion, frequent occlusions, and cluttered backgrounds. While SportsMOT is primarily oriented towards player tracking, it provides a useful reference for understanding the tracking challenges specific to indoor sports venues.

2.3.2 Kalman Filtering and Kinematic Motion Models

The Kalman filter (Kalman, 1960) is a widely used recursive state estimator that maintains a probability distribution over the hidden state of a dynamical system and updates this distribution as new measurements become available. In the context of object tracking, the hidden state typically comprises position and velocity, and the measurement is the detected bounding box centre. The filter operates in two alternating steps: a prediction step, which propagates the state estimate forward in time using a kinematic motion model; and an update step, which corrects the prediction using the new observation, weighted by the relative uncertainty of the model and the measurement.

The choice of motion model has a significant impact on tracking accuracy. A constant-velocity model is appropriate when the inter-frame interval is short and the object's acceleration is small relative to measurement noise; a constant-acceleration model better handles objects subject to gravity, such as balls in flight. In sports tracking, the Kalman filter is frequently combined with the tracking-by-detection paradigm to handle gap frames — frames in which no detection is available — by propagating the state estimate forward without an update, thereby interpolating the trajectory across short occlusions without breaking trajectory continuity (Bewley et al., 2016).

2.3.3 False Positive Suppression and Physical Plausibility Constraints

In environments with visually complex backgrounds, detectors frequently produce false positive detections that, if accepted by a tracker, corrupt the trajectory. Several complementary strategies have been proposed for suppression.

Background subtraction methods, including the Mixture of Gaussians model (MOG2) (Zivkovic, 2004), model the expected appearance of each pixel under background-only conditions and flag pixels that deviate significantly as foreground. In the tracking context, foreground masks can serve as auxiliary validation signals: a detection that does not coincide with foreground activity is more likely to be a false positive on a static background element. This approach has been applied in sports tracking to reduce the false positive rate on non-moving scene elements such as court markings, signage, and wall decorations (Kamble et al., 2019).

Physical plausibility constraints provide a complementary layer of validation. Because a real ball cannot instantaneously teleport between distant positions, any candidate detection implying a displacement exceeding a physically plausible bound — derived from the known maximum speed of the ball and the camera frame rate — can be rejected as a spurious outlier. This form of speed-gated rejection is computationally inexpensive and robust to a wide class of false positives that appear in spatially unrelated regions of the frame (Kamble et al., 2019).

Together, these mechanisms reflect a broader principle in sports tracking: exploiting domain knowledge about the physical behaviour of the tracked object to constrain and validate the output of learned detectors operating on challenging footage.

2.4 Scoreboard Reading in Sports Video

In broadcast and fixed-camera recordings of volleyball matches, the on-screen scoreboard provides a continuous record of the match state: the current score for each team, the set count, and related information. Automatic reading of this information can serve as a contextual signal for event-level analysis, as a score change of $+1$ for either team is a reliable indicator of the end of a rally and the attribution of a point.

2.4.1 Scene Text Recognition in Sports Broadcasts

The automatic extraction of text from natural images and video — scene text recognition (STR) — is a well-studied area of computer vision, with applications ranging from document digitisation to augmented reality (Long et al., 2021). In the sports broadcast context, however, the problem is considerably more constrained than the general case. The scoreboard occupies a fixed or predictable region of the frame, the character set is restricted to decimal digits and short numbers, and the rendering style is consistent within a single broadcast. These constraints allow simpler and more targeted recognition approaches to be effective where general-purpose scene text recognisers might be unnecessarily complex.

Two broad approaches to digit recognition in sports scoreboards have been explored in the literature. Template matching approaches compare a cropped digit region against a library of reference images using normalised cross-correlation or similar similarity measures. These methods are computationally inexpensive, require no labelled training data beyond the reference templates, and are robust to the controlled variation present in a single broadcast. Their main limitation is sensitivity to scale, font rendering differences, and contrast inversion (Guo et al., 2011). Convolutional classifier approaches train a small convolutional network to map a cropped digit image to a class label, and have demonstrated high accuracy in digit recognition benchmarks (LeCun et al., 1998). In practice, the stability of CNN-based classifiers in end-to-end scoreboard pipelines depends critically on the reliability of the digit segmentation step that precedes classification; errors in segmentation propagate directly to recognition errors.

2.4.2 Temporal Validation and Rule-Based Consistency

A fundamental challenge in scoreboard reading from video is the noisiness of per-frame readings: individual frames may produce incorrect digit readings due to compression artefacts, interlacing, motion blur, or brief illumination changes. A naïve approach that accepts each per-frame reading directly will generate spurious score changes, leading to incorrect event boundaries.

Temporal validation addresses this by requiring consistent readings across multiple consecutive frames before accepting a score change (Guo et al., 2011). This multi-frame confirmation mechanism reduces the impact of transient frame-level noise on downstream analysis. A complementary strategy is the application of domain rule constraints: in volleyball, the score for either team can only increase by exactly one per rally, set scores are bounded by the rules of the game, and scores within a set cannot decrease. Readings that violate these constraints are necessarily erroneous and can be discarded or corrected without requiring additional ground truth. The combination of temporal consistency checking and domain rule enforcement represents a well-established approach to robust reading of constrained text in structured domains (Guo et al., 2011).

2.5 Datasets for Volleyball and Sports Vision

The availability of annotated datasets is a fundamental prerequisite for the development and evaluation of data-driven computer vision systems. In the context of volleyball video analysis, the landscape of publicly available datasets is somewhat fragmented: existing resources address individual sub-tasks (activity recognition, player tracking, ball detection) but few cover the full combination of ball trajectory, event labelling, and contextual match state that a complete match analysis pipeline requires.

2.5.1 Activity Recognition Datasets

The Volleyball Dataset introduced by Ibrahim et al. (2016) contains annotated video clips from broadcast volleyball matches and has been widely used as a benchmark for group activity recognition. Clips are labelled with both individual player actions and collective team activities, enabling the evaluation of hierarchical recognition models. The dataset has been influential in establishing a standard benchmark for relational action recognition in team sports. Its limitation for the purposes of this dissertation is that it provides activity labels at the clip level, without ball position annotations or frame-level event boundaries.

2.5.2 Sports Detection and Tracking Datasets

The DeepSportradar-v1 dataset (Giancola et al., 2022), introduced alongside a challenge at the 2022 ACM Multimedia conference, provides high-resolution sports video annotations for tasks such as ball localisation, player segmentation, camera calibration, and player re-identification. It represents a significant contribution by consolidating diverse sports footage under a common annotation format, facilitating cross-sport and cross-task evaluation.

The SportsMOT dataset (Cui et al., 2023), presented at ICCV 2023, is specifically designed for multi-object tracking in sports. It contains approximately 240 video sequences across basketball, volleyball, and football, with dense bounding box annotations and persistent identity labels in the MOT Challenge format. The volleyball sequences in SportsMOT are particularly relevant because they were recorded in indoor venues under conditions representative of competitive play, including artificial lighting, complex backgrounds, and fast-moving players. SportsMOT provides a useful reference for understanding the tracking challenges of the indoor volleyball domain, though its primary focus is on player tracking rather than ball trajectory analysis.

Several volleyball-specific ball detection datasets have been made available through platforms such as Roboflow, typically constructed by community contributors from publicly available match footage. These datasets vary considerably in annotation quality, venue diversity, and coverage of challenging conditions such as motion blur and occlusion, and have not been systematically benchmarked in the peer-reviewed literature.

2.5.3 Limitations of Existing Datasets

A consistent limitation of the available datasets for volleyball analysis is their lack of coverage for the specific combination of tasks addressed in this dissertation: ball detection in single-camera indoor footage, temporal trajectory reconstruction, scoreboard reading, and event inference at the rally level. Existing datasets either address activity recognition at the action clip level without ball position annotations, or provide ball tracking annotations without event labels or scoreboard information. The absence of a dataset that covers all four components simultaneously reflects a broader gap in the literature, which this dissertation addresses through the construction of a custom annotated dataset described in Chapter 3.

Table 2.1: Dataset coverage in related work.

Dataset/source	**Main focus**	**Limitation for this dissertation**
Volleyball Dataset (Ibrahim et al., 2016)	Group activity recognition from broadcast clips	Provides activity labels but not ball positions, scoreboard readings, or rally-level trajectory records.
DeepSportradar-v1 (Giancola et al., 2022)	Multi-sport detection and event understanding	Useful for sports vision benchmarking, but not tailored to the complete volleyball pipeline required here.
SportsMOT (Cui et al., 2023)	Multi-object player tracking in sports scenes	Includes volleyball sequences but focuses on player identities rather than ball trajectory and scoreboard state.
Community ball datasets	Volleyball ball detection examples	Annotation quality, venue diversity, and coverage of blur or occlusion are inconsistent and not systematically benchmarked.
Custom dataset in this work	Ball detection in the target indoor venue	Supports the proposed detector but remains venue-specific and does not by itself solve scoreboard reading or event labelling.

2.6 Summary and Positioning of this Dissertation

The preceding sections have reviewed the literature across five areas relevant to automatic volleyball match analysis. The following observations motivate the approach taken in this dissertation.

First, the literature on volleyball performance analysis confirms that event-level statistics are practically valuable but that fully automatic extraction remains challenging. Existing automatic systems either address activity recognition without direct ball tracking, or perform ball tracking without event-level interpretation (Ibrahim et al., 2016); (Chen and Chu, 2021); (Huang, 2023).

Second, single-stage detectors from the YOLO family have become the practical standard for real-time or near-real-time sports object detection, offering a favourable trade-off between accuracy and throughput for ball and player detection (Jocher et al., 2024). Small object detection in volleyball footage presents specific challenges — ball size, motion blur, and venue-specific false positives — that generic pretrained models do not address adequately, motivating domain-specific fine-tuning and hard negative training.

Third, per-frame detection is insufficient for reliable trajectory analysis. Temporal continuity must be maintained across occlusion gaps, and false positives must be suppressed using a combination of kinematic prediction, motion-based validation, and physical plausibility constraints (Kalman, 1960); (Zivkovic, 2004); (Kamble et al., 2019). The tracking-by-detection paradigm, augmented with these domain-specific constraints, provides an appropriate framework for single-ball tracking in volleyball video.

Fourth, automatic scoreboard reading from sports video can be achieved through constrained digit recognition combined with temporal validation and rule-based consistency checks. The constrained nature of the scoreboard domain — fixed position, limited character set, consistent rendering — means that targeted template-based approaches can outperform general-purpose scene text recognisers in practice, provided that temporal validation is applied to suppress noisy per-frame readings (Guo et al., 2011).

Fifth, no existing public dataset covers the specific combination of ball detection, temporal tracking, scoreboard reading, and event inference required for the system developed in this dissertation. Available datasets address individual sub-tasks but not their integration.

Table 2.2 summarises key characteristics of the most directly relevant related works and positions this dissertation relative to them.

Table 2.2: Related work comparison.

Work	Ball Det.	Tracking	Scoreboard	Events
Ibrahim et al. (2016)	–	–	–	✓
Huang (2023)	–	GCN	–	✓
This work	**YOLOv8**	**Kalman**	✓	✓

To the best of our knowledge, no published system combines all four of the components addressed in this dissertation — ball detection, temporal tracking, automatic scoreboard reading, and event inference — into a single pipeline operating on uncontrolled single-camera match footage. Prior work has addressed these components in isolation or in partial combination, as Table 2.2 illustrates. This dissertation addresses a practical integration gap: bringing together these components in a hybrid pipeline that is robust to the visual conditions of a real indoor volleyball venue and produces structured statistical output directly usable by coaching staff.

The approach taken is deliberately hybrid rather than end-to-end learned: each component is designed around the specific challenges and constraints of its sub-task, using established techniques — YOLOv8 fine-tuning, Kalman-assisted tracking, template-based digit recognition — augmented with domain knowledge about the physical and rule-based structure of volleyball. This design choice reflects both the limited availability of labelled volleyball data for end-to-end training and the interpretability requirements of a coaching-support application. The system is not described as fully automatic, as calibration and scoreboard region-of-interest selection require a one-time manual setup step.

The next Chapter describes the system design and application scenarios, showing how the components reviewed here are integrated into a coherent processing pipeline.

Chapter 3

System Specification and Application Scenario

This Chapter specifies the application context, target footage, system requirements, supporting dataset and assets, high-level architecture, and expected outputs before the implementation details are discussed in the following Chapter.

3.1 Application Scenario

The system is evaluated on fixed-camera indoor volleyball footage from two real venues: *Pav. Arq. Jerónimo Reis Espinho* and *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. The pipeline remains venue-dependent, since calibration, scoreboard templates, and detector behaviour may need to be adjusted for each recording setup. The intended users are coaching staff or technical analysts at club or semi-professional level who wish to extract structured rally and event statistics from a recorded match without investing in multi-camera infrastructure or commercial performance-analysis platforms (Cheng et al., 2022).

The workflow is offline and semi-automatic. The operator performs a short manual initialisation at startup, after which the system processes the selected video segment automatically, frame by frame. The active processing window is defined by a configurable start and end timestamp (`start_ts` and `end_ts`), so analysis can be focused on the portion of the recording that is relevant to the session rather than requiring the entire raw file to be processed in a single pass.

The design reflects a deliberate trade-off: a small one-time setup cost is accepted in exchange for automatic processing during the analysis stage. This makes the system practical for repeated use across matches recorded under similar conditions, provided that venue-specific assets are checked or adjusted when the recording setup changes.

The main application sub-scenarios derived from this workflow are described in Section 3.6, including ball detection and tracking, scoreboard reading and rally segmentation, court calibration, and event classification.

3.2 Target Footage and Technical Challenges

The target footage is a single fixed-camera recording of a competitive volleyball match. The camera is positioned at an elevated, lateral angle that provides a full view of the court. This perspective is standard for club-level recordings and offers good coverage of ball trajectories, player positions, and the on-screen scoreboard, but it introduces a number of technical challenges (Fujii, 2025).

Most development and parameter tuning were carried out using footage from *Pav. Arq. Jerónimo Reis Espinho*. However, the screenshots and walkthrough examples shown in this part of the report were generated from the active footage at the time of documentation, recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. That source video is approximately 103 minutes long, recorded at 1280×720 pixels and 30 frames per second. The active segment shown in this part of the report corresponds to the first set of the match, which occurs between 00:26:00 and 00:46:27 within the source recording.

The main technical challenges present in this footage are:

- **Small target size**: the ball occupies very few pixels in most frames, particularly when play is on the far side of the court. This places ball detection firmly in the small-object regime (Tong et al., 2020).
- **Motion blur**: during fast attacks and serves, the ball moves quickly enough to appear as a blurred streak rather than a compact circular object, making frame-by-frame detection unreliable.
- **Occlusions**: players, the net, the referee stand, and advertising banners can hide the ball for one or more consecutive frames. The tracking module must bridge these gaps without losing trajectory continuity.
- **Complex background**: the venue contains repeated visual patterns on the rear wall and floor that can generate false-positive detections when using a generic pretrained detector. Targeted mitigation is required.
- **Scoreboard overlay**: the on-screen scoreboard is a visually dense region that must be read reliably. At the same time, it provides a useful temporal signal because a score change indicates the end of a rally.

3.3 System Requirements

3.3.1 Functional Requirements

The system must fulfil the functional requirements listed in Table 3.1.

Table 3.1: Functional requirements.

ID	Requirement
FR1	Detect the volleyball in each frame of the selected video segment.
FR2	Track the ball over time, maintaining a coherent trajectory across frames and bridging short detection gaps.
FR3	Read the on-screen scoreboard from a fixed region of interest within the frame.
FR4	Identify rally endings by detecting and validating score changes.
FR5	Classify the technical event associated with each completed rally.
FR6	Export the analysis results in machine-readable formats suitable for further inspection.

3.3.2 Non-Functional Requirements

Table 3.2 summarises the non-functional requirements that constrain the prototype design.

Table 3.2: Non-functional requirements.

ID	Requirement
NFR1	The system processes pre-recorded video segments and is not designed for live broadcast use.
NFR2	A short manual initialisation step is required at the beginning of each run, but no human intervention is needed during the processing stage.
NFR3	Detection, tracking, scoreboard reading, and event classification are implemented as separate components that can be evaluated and replaced independently.
NFR4	The system stores calibration data, runtime parameters, and local assets so that an analysis can be repeated under the same configuration, although formal repeatability testing across multiple runs is outside the scope of this dissertation.
NFR5	The system operates on footage from a single consumer-grade fixed camera, without multi-camera synchronisation or specialised hardware.
NFR6	The pipeline must tolerate short occlusions, false-positive detections from the venue background, and noisy per-frame scoreboard readings.

These requirements define the operational scope of the prototype and are used to guide the implementation choices described in Chapter 4.

3.3.3 Manual Setup Requirements

Two manual steps are required at startup before automatic processing can begin:

1. **Court calibration**: the operator clicks six reference points in a representative frame — the four corners of the court and two points on the top of the net. The system uses these correspondences to compute a homography from image coordinates to real court coordinates and to define the net geometry (Magera et al., 2024).

2. **Scoreboard region-of-interest selection**: the operator interactively selects the scoreboard region in the first processed frame. The system then refines and stores this region for use throughout the selected segment.

These steps are performed once per video setup. The resulting calibration data is saved to a file and can be reused for subsequent analyses of footage recorded from the same camera position. Automatic court calibration and automatic scoreboard discovery are not part of the current pipeline.

In addition to the interactive steps, the pipeline depends on a set of local assets that must be present before execution: a trained ball detection model, a library of digit templates for scoreboard reading, and the source video file. The current configuration is tailored to the selected recording setup and is not yet designed for generic multi-venue use without venue-specific calibration and asset updates.

3.4 Dataset Specification

3.4.1 Ball Detection Dataset

The ball detector is trained on a custom single-class dataset extracted from the development footage recorded at *Pav. Arq. Jerónimo Reis Espinho*. The dataset contains 600 images in total, divided into 480 training images (80.0%) and 120 validation images (20.0%). All annotations follow the YOLO format, where each label file specifies a single class, `ball`, together with the normalised bounding-box centre coordinates and dimensions. Of the 600 images, 476 contain positive ball annotations, while 124 images (20.7%) are background samples with empty label files, distributed as 95 background samples in the training split and 29 in the validation split. These negative examples help the detector learn to reject venue-specific false positives caused by repeated visual patterns on the rear wall, floor markings, and other background distractors. A consistency check of the dataset structure found no missing label files and no extra label files without a corresponding image. Figure 3.1 shows representative positive and background samples, while Figure 3.2 summarises the training/validation split and label distribution.

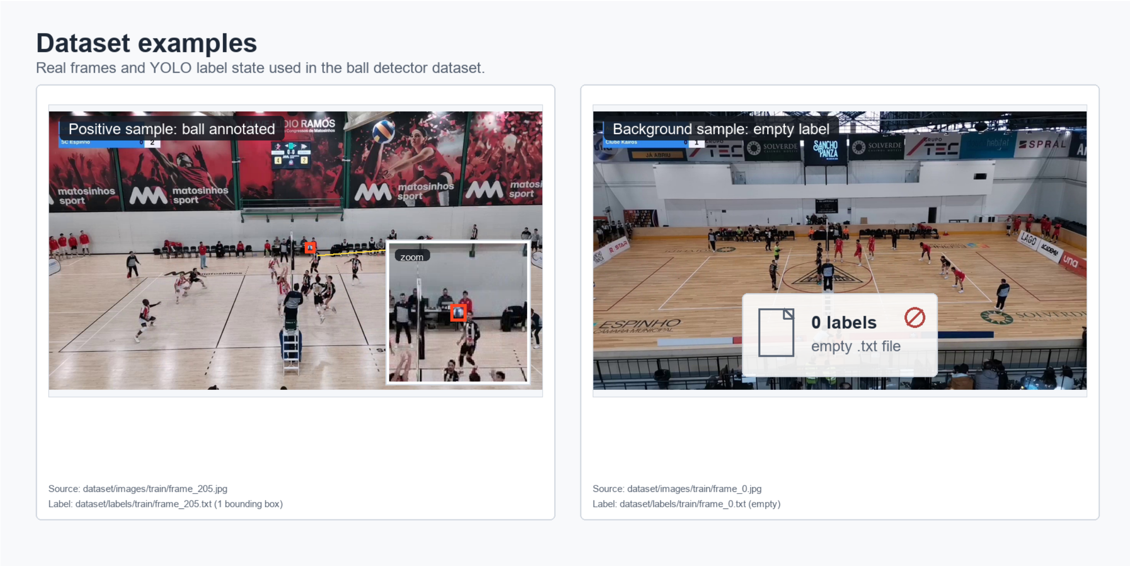


Figure 3.1: Representative samples from the custom ball detection dataset, including annotated volleyball instances and background-only frames used to reduce venue-specific false positives.

Because the images are extracted from a continuous video recording, temporally adjacent frames tend to be visually similar. The dataset construction procedure therefore uses an ordered and reproducible split strategy intended to reduce the risk of inflated validation metrics caused by near-duplicate adjacent frames.

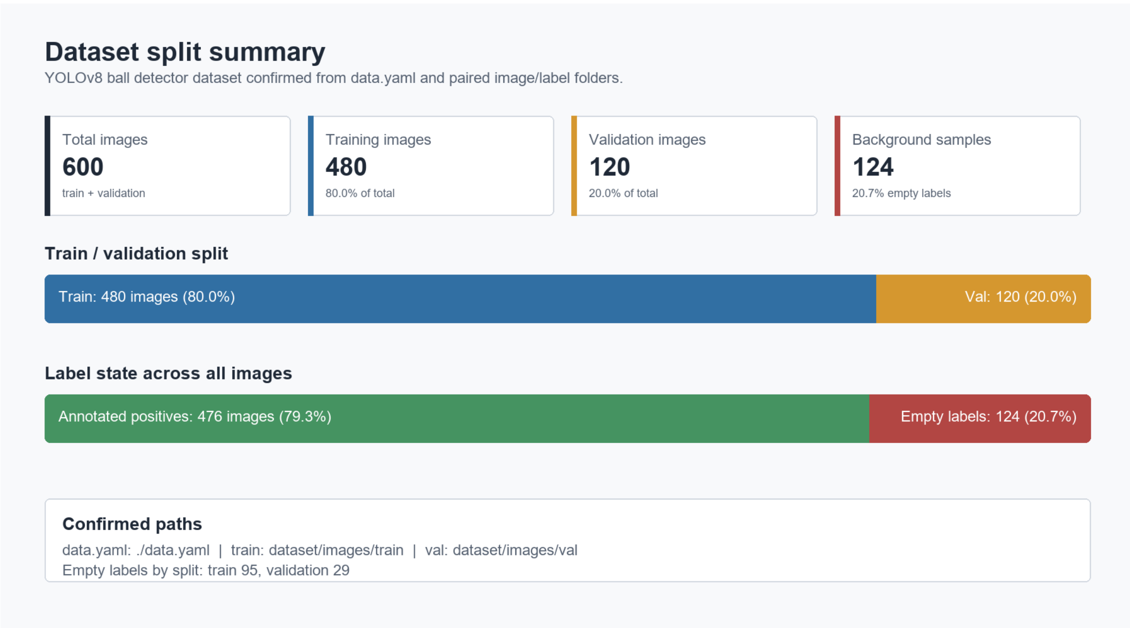


Figure 3.2: Summary of the custom dataset split, showing the distribution of annotated ball images and background-only samples across the training and validation subsets.

3.4.2 Scoreboard Template Assets

The scoreboard reading module uses a small library of digit reference images covering the digits zero through nine. These templates were prepared for the scoreboard style used during development at *Pav. Arq. Jerónimo Reis Espinho* and capture the specific font, size, and rendering style of that scoreboard. The templates are stored as separate image files and loaded at runtime by the scoreboard reader. This approach is domain-specific by design: it yields stable recognition for the target scoreboard style without requiring a general-purpose scene text recogniser.

3.4.3 Dataset Limitations

The ball detection dataset is intentionally small and venue-specific. It was constructed from a single match recording at *Pav. Arq. Jerónimo Reis Espinho*, which means that the trained detector is expected to generalise well to footage from the same setting but may require retraining or fine-tuning when applied to footage from different venues or camera positions. Similarly, the digit templates are tied to the scoreboard style used during development and would need to be replaced or supplemented for use with a different scoreboard format.

The current dataset documentation does not include an explicit verification that the split is fully free of temporal leakage. Although the split procedure is reproducible and ordered, the absence of a formal leakage analysis is acknowledged as a limitation.

At the time of writing, no quantitative benchmark evaluation against a larger independently annotated ground truth has been performed for the full pipeline.

3.5 System Architecture

3.5.1 Pipeline Overview

The system follows a modular sequential pipeline. Starting from the manually configured video segment, it processes frames in order and accumulates evidence across detection, tracking, and scoreboard reading until a rally boundary is confirmed. At that point, the rally is classified and its record is written to the output files.

Figure 3.3 presents the main system components and the transition from the initial manual setup to automated processing and structured output generation.

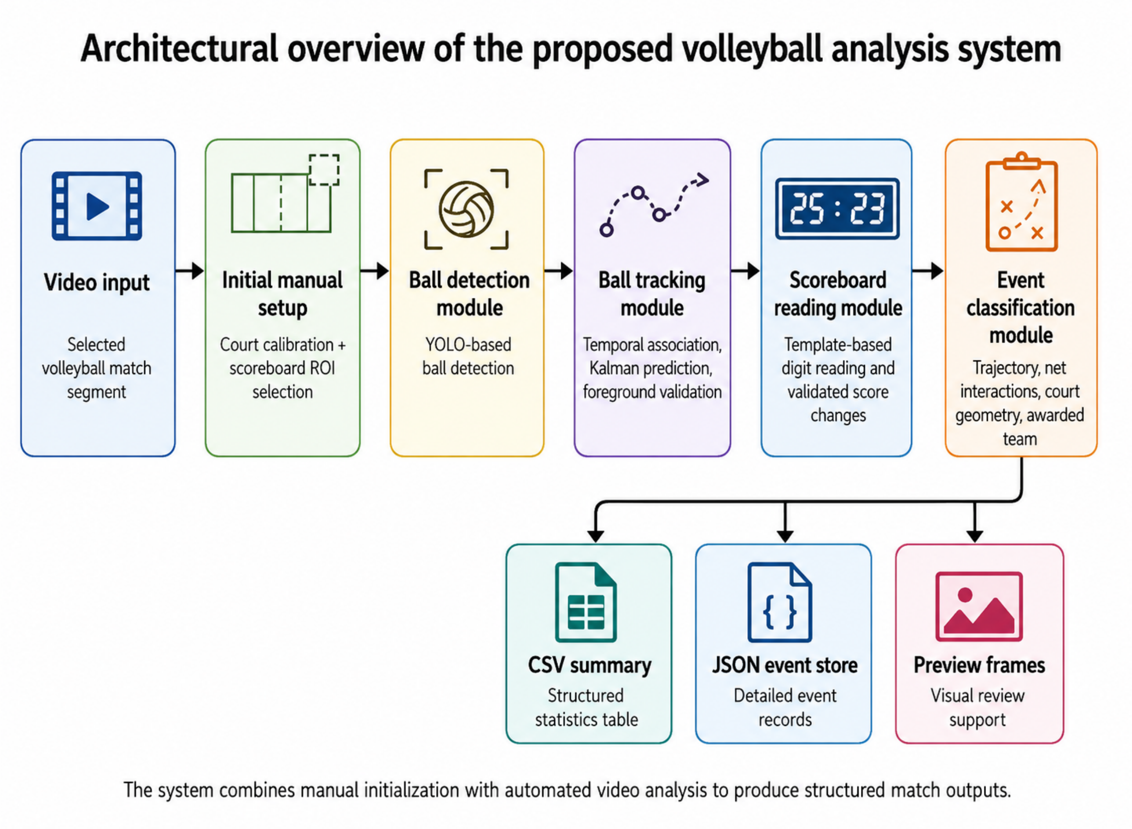


Figure 3.3: Architectural overview of the proposed volleyball analysis system. The system combines an initial manual setup stage with automated video processing modules for ball detection, ball tracking, scoreboard reading, event classification, and structured output generation.

The system begins with a selected volleyball video segment. An initial manual stage establishes the court geometry and net position through calibration and defines the scoreboard region of interest. The remaining segment is then processed automatically. The detection module identifies candidate ball observations, while the tracking module associates them over time to construct a coherent trajectory. In parallel, the scoreboard reader validates score changes. A valid change confirms the end of a rally and identifies the team awarded the point. The event classifier then combines the reconstructed trajectory, calibrated court geometry, net-related evidence, and scoreboard context to infer the final rally-ending event. The resulting records are stored as CSV summaries, detailed JSON events, and annotated preview frames for visual inspection.

The main processing stages are summarised in Table 3.3.

Table 3.3: Pipeline stages.

Stage	Role in the pipeline
Startup and calibration	Establishes court geometry and allows the operator to select the scoreboard region of interest.
Frame extraction	Decodes frames between `start_ts` and `end_ts` and prepares them for downstream modules.
Ball detection	Produces candidate ball observations with confidence scores and bounding-box coordinates.
Ball tracking	Links detections over time into a coherent trajectory and bridges short occlusion gaps.
Scoreboard reading	Samples the scoreboard, stabilises digit readings through voting, and validates score progression.
Rally segmentation	Uses confirmed score changes to close rallies and trigger event analysis.
Event classification	Combines trajectory, court geometry, and score context to assign the rally label.
Output generation	Writes CSV summaries, JSON event records, and event preview images.

3.5.2 Design Constraints and Assumptions

The architecture is built around three explicit assumptions that shape all design decisions:

- The camera is fixed throughout the processed segment. No camera motion compensation is applied.
- The scoreboard is visible in a fixed region of the frame for the entire segment. The system does not attempt to locate the scoreboard automatically.
- The court geometry does not change between the calibration step and the end of the processed segment. The same homography is applied throughout.

The pipeline is designed as a research prototype rather than a production system. It has not been benchmarked against a commercial platform or evaluated on footage from multiple venues.

3.5.3 Input and Output Interfaces

The primary inputs to the system are:

- The source video file, accessed via the path configured in a config file `config.py`.

- The processing time window, defined by `start_ts` and `end_ts`.
- The calibration file produced during the interactive setup step.
- The ball detection model weights and the digit template library.

The primary outputs are described in detail in Section 3.7.

3.6 Application Sub-Scenarios

3.6.1 Ball Detection and Tracking

Ball detection and tracking are the core visual components of the pipeline. The goal is not merely to detect the ball in isolated frames but to maintain a stable and coherent trajectory across the full duration of a rally. This trajectory is the primary input to the event classification stage: the system reasons about where the ball was, how it moved relative to the net, and how quickly it travelled in order to infer what type of action concluded the rally.

Because volleyball is a fast sport with frequent occlusions and a small ball, frame-by-frame detection is insufficient on its own. The tracking module must bridge short detection gaps, suppress false positives arising from background distractors, and enforce physical plausibility constraints on the accepted trajectory.

3.6.2 Scoreboard Reading and Rally Segmentation

Scoreboard reading serves as the primary temporal synchronisation mechanism in the pipeline. The system monitors the score displayed on the broadcast overlay and confirms a rally boundary when it detects a validated $+1$ increment for either team. This approach avoids the need for a trained end-of-rally classifier, because the score change itself is a reliable observable signal directly embedded in the video.

The system does not attempt generic text recognition. It is calibrated specifically to the scoreboard format represented by the domain-specific digit templates prepared during development at *Pav. Arq. Jerónimo Reis Espinho*, as described in Section 3.4.2. Raw readings are stabilised through a temporal voting mechanism and validated against the logical constraints of the volleyball scoring system before a score change is accepted as a rally boundary (Guo et al., 2011).

3.6.3 Court Calibration

Court calibration translates image-space coordinates into a normalised court coordinate system that allows the rest of the pipeline to reason about spatial relationships in physical terms. The operator identifies six points in a representative frame: the four corners of the court and two points on the top of the net. A homography is estimated from these correspondences and used throughout the segment to project ball positions from image space to court space (Magera et al., 2024).

The net geometry, derived from the two clicked net points, defines the court halves and the net proximity zone used by the event classifier. All spatial reasoning in the event classification stage — including net crossing detection, out-of-bounds judgement, and block proximity — is performed in court coordinates rather than directly in image pixels.

3.6.4 Event Classification

The event classification stage assigns a label to each completed rally based on the ball trajectory, the court geometry, and the identity of the team that gained the point according to the scoreboard. Classification is heuristic and geometry-driven; the system does not use an end-to-end action recognition model trained on a labelled event dataset.

The system defines an internal taxonomy of event categories. At the time of writing, the categories observed in the operational output are `attack`, `block`, `ball_out`, `ball_on_net`, and `error`. The codebase also includes additional internal labels, including `ace`, `freeball`, `spike`, and `undefined`. In the evaluated version of the prototype, these labels are consolidated where appropriate. Spike-like and freeball-like outcomes are reported under the broader `attack` category, reducing fragmentation in the output statistics and making the evaluation more consistent given the limited number of reviewed rallies. The `undefined` label is reserved for internal fallback cases and is not treated as a final event category in the reported evaluation. Each classified event is stored together with a confidence value, a reason string, and a notes field that traces the evidence used in the classification decision.

3.7 Output Specification

The pipeline produces three output artefacts:

1. **Rally summary**: a tabular file containing one row per detected rally. Each row records the rally identifier, start and end timestamps, rally duration, the identified attacker, the type of point, the winning team, the number of net crossings, and ball speed statistics (mean and maximum) for that rally.

2. **Event log**: a structured JSON file containing one record per detected event. Each record includes the event type, category, point type, start and end frame indices, timestamps, team and side assignments, confidence, reason, notes, and a path to a

representative preview image. This file is the primary machine-readable output of the pipeline and supports detailed per-event inspection.

3. **Event preview images**: one annotated frame per event, saved as a visual reference that allows the operator to inspect the field state at the moment the event was classified.

An additional file json, is generated when the pipeline is invoked in evaluation mode. This file contains aggregate counts of the main event categories, team point totals, and a final score snapshot. It is intended for development and benchmarking purposes and is not produced during standard operation.

The specification above defines the application scenario, the technical challenges of the target footage, the functional and non-functional requirements, the manual setup steps, the dataset and template assets, the high-level architecture, and the intended outputs of the system. The system is a semi-automatic offline pipeline that combines a short interactive initialisation with automatic frame-by-frame processing over a user-selected video segment. It is designed for club-level post-match analysis across fixed-camera indoor recordings, but remains venue-dependent and is not claimed to be fully automatic, real-time, or immediately generalisable to other venues.

The following Chapter describes how this specification is realised in code, covering the concrete calibration workflow, the ball detection and tracking logic, the scoreboard reading implementation, and the event classification routines in detail.

Chapter 4

Implementation

This Chapter describes how the system specification defined in Chapter 3 was realised in code. It covers the development environment, the pipeline entry point, the calibration procedure, the ball detection and tracking logic, the block detector, the scoreboard reader, the rally management and event classification stage, and the output generation mechanisms. The presentation follows the active runtime pipeline; support scripts and experimental branches are identified as such where they arise. Quantitative evaluation of the implemented components is reserved for Chapter 5.

4.1 Implementation Overview

The system is organised as an offline, modular pipeline that transforms a locally stored video segment into structured match statistics. Execution begins in `main.py`, which orchestrates the full sequence: directory setup, court calibration, scoreboard region selection, frame-by-frame analysis, and output persistence. Each major concern — detection, tracking, scoreboard reading, court geometry, rally management, event classification, and output writing — is encapsulated in a dedicated module, so that components can be developed, tested, and replaced independently without modifying the rest of the pipeline.

The pipeline is file-based and non-interactive during the analysis stage. It reads frames from the configured video segment, accumulates evidence across detection, tracking, and scoreboard readings, and finalises each rally when a confirmed score change is received. All outputs are written to a local output directory upon rally finalisation or at the end of the run.

Table 4.1: Implementation modules.

Module	Responsibility
Configuration and entry point	Defines runtime paths, model settings, thresholds, time windows, and execution mode.
Court calibration	Stores the homography and net reference points used by the geometric reasoning components.
Ball detection	Runs the fine-tuned YOLOv8 model and emits candidate ball bounding boxes.
Tracking and validation	Maintains the Kalman state, rejects implausible detections, and recovers short gaps.
Scoreboard reading	Extracts score digits from the selected region and validates changes through voting and volleyball rules.
Rally and event logic	Converts score changes and trajectories into rally records and event labels.
Output generation	Persists CSV, JSON, statistics summaries, and event preview images.

4.2 Development Environment and Runtime Dependencies

The implementation is written in Python and targets a local desktop environment with optional CUDA acceleration. The principal runtime dependencies are OpenCV for image processing, background subtraction, and template matching; Ultralytics YOLOv8 (Jocher et al., 2024) for ball detection; PyTorch (Paszke et al., 2019) as the underlying deep learning runtime; NumPy (Harris et al., 2020) for numerical computation; and Pandas (McKinney, 2010) for rally-summary serialisation.

The pipeline can operate in two modes. In the default visual mode, an analysis overlay is rendered and displayed in real time during processing. In headless or evaluation mode, activated with the `--eval` flag, the display window is suppressed and the system runs without a graphical interface, printing periodic progress messages instead. Both modes require the same manual initialisation steps; `--eval` does not bypass calibration or region selection.

4.3 Pipeline Entry Point and Configuration

`main.py` is the single entry point for a standard analysis run. On startup it calls a directory-setup routine to ensure that the output, calibration, and preview directories exist, then immediately invokes court calibration. After calibration returns, the scoreboard reader is instantiated, the analysis video is opened, and the operator selects the scoreboard region of interest in the first processed frame. The frame loop then begins and runs until the end of the configured time window.

All runtime behaviour is controlled through `config.py`, which functions as the central configuration surface. It encodes the video file path, the processing time window (`start_ts` and `end_ts`), the model checkpoint path, detection confidence thresholds, Kalman filter noise parameters, scoreboard voting parameters, output paths, and mode flags. Because the configuration is a single Python object that is passed to every major module, changing a threshold or switching between model variants requires editing only one file.

4.4 Court Calibration Implementation

Court calibration translates image-space pixel coordinates into metric court coordinates so that downstream modules can reason about spatial relationships in physical terms. The calibration routine opens the source video, seeks to a representative frame near the beginning of the recording, and prompts the operator to click six reference points: the four corners of the court in order, followed by two points on the top of the net.

From the four court corners, a homography matrix $\mathbf{H}$ is estimated using `cv2.findHomography` with RANSAC, mapping the clicked image points to the known real-world positions of a standard volleyball court modelled as a 9 m $\times$ 18 m rectangle (Magera et al., 2024). The two net points define a net line segment in image space, which is stored alongside $\mathbf{H}$ and used by the court geometry module to determine court halves, net proximity, and crossing events.

The resulting calibration data — the homography matrix, the net line, and the scoreboard region identified during the subsequent selection step — are persisted in `calibration/field_params.json`. This file is loaded at the start of each subsequent run, so the operator need not repeat the calibration step for analyses of the same footage recorded from the same camera position.

Figure 4.1 illustrates this manual calibration step and the selected court and net reference points.

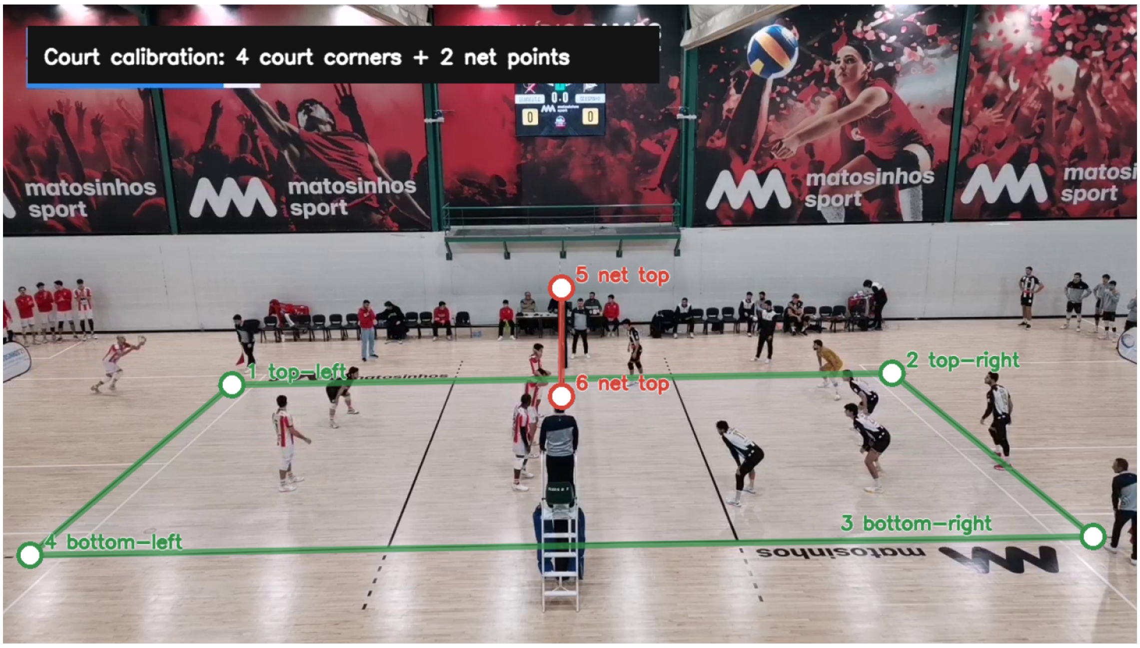


Figure 4.1: Example of the manual court calibration stage, showing the selected court corners and top-of-net reference points used to establish the homographic court coordinate system.

The `court_geometry.py` module wraps the calibrated homography and net line and exposes helper functions used throughout the pipeline: projecting a pixel position to court coordinates, checking whether a point is inside the court boundaries, determining which side of the net a position belongs to, testing whether a line segment intersects the net zone, and classifying a position as belonging to a named court zone.

4.5 Ball Detection Implementation

4.5.1 Dataset Preparation

The detector was trained using the custom YOLO-format dataset specified in Section 3.4.1. This implementation section therefore focuses on how the dataset is organised, ingested, and used by the training pipeline, rather than repeating the dataset specification in full. The dataset is organised in the Ultralytics directory layout, with image and label subdirectories for training and validation splits. Labels follow the YOLO format, where each annotation specifies the class index (always zero, for `ball`) and the normalised bounding-box centre and dimensions.

The corresponding `data.yaml` file points the Ultralytics trainer to `dataset/images/train` and `dataset/images/val`, while the matching label directories are inferred from the standard YOLO folder structure.

Dataset assembly is supported by `ingest_dataset.py`, which collects source frames and their YOLO labels, applies a reproducible random split with a fixed seed, and copies the resulting files into the training and validation directories. Because the images originate from a continuous video recording, the split procedure is designed to keep the evaluation subset temporally separated from the training subset, reducing the risk of inflated validation metrics from near-duplicate adjacent frames.

4.5.2 YOLOv8 Training

The detector was trained using the Ultralytics training pipeline, starting from the `yolov8s` base model pretrained on COCO (Lin et al., 2014). Fine-tuning from pretrained weights is preferred over training from scratch because the backbone's generic low-level feature detectors — edges, corners, and texture responses — transfer well to the ball detection task, while the custom training data adapts the detection head to the specific appearance of the volleyball and the background of *Pav. Arq. Jerónimo Reis Espinho*. Training was performed at an input resolution of 1280×720 pixels to preserve spatial detail for the small ball target.

The active checkpoint used at runtime is the best-validation-loss model from the final training run. The model is a single-class detector, and all generic COCO classes are absent from its output.

4.5.3 Runtime Inference

At runtime, `tracker.py` invokes the YOLO detector on each processed frame. The frame is rescaled to the configured inference width before being passed to the model, and the resulting bounding box predictions are filtered by an adaptive confidence threshold before being forwarded to the ball-tracking core. The runtime code also supports a fallback to a generic YOLOv8 model using the COCO sports ball class, which allows the system to operate in the absence of the custom checkpoint, though with reduced specificity.

Raw detector outputs are not accepted directly as ball positions. They are instead treated as candidate observations that must pass a further multi-criterion validation stage, described in the following section.

4.6 Ball Tracking Implementation

4.6.1 Kalman-Based Tracking

Ball motion between frames is modelled with a discrete constant-velocity Kalman filter (Kalman, 1960) implemented using OpenCV's `KalmanFilter` class. The state vector comprises the ball's two-dimensional pixel position and its instantaneous velocity: $\mathbf{x} = [x, y, v_x, v_y]^\top$. The observation vector comprises the pixel position alone: $\mathbf{z} = [x, y]^\top$.

At each frame, the filter first executes a prediction step, propagating the current state estimate forward by one time step using the constant-velocity transition model. If a valid candidate detection is available after the validation stage, the filter executes an update step that corrects the prediction using the observation, weighted by the configured process and measurement noise covariances. If no valid candidate is available, only the prediction step is executed, and the predicted position is used for gap handling as described in Section 4.6.4.

4.6.2 Foreground-Based Validation

The ball tracking core (`ball_tracking_core.py`) does not rely on YOLO confidence alone to accept or reject a candidate detection. Each candidate is scored against multiple criteria, of which foreground support is one of the most discriminative. A foreground mask is computed at runtime using the Mixture of Gaussians background subtractor MOG2 (Zivkovic, 2004). A candidate whose bounding box coincides with a region of significant foreground activity is assigned a higher foreground-support score than one that lies in a static area of the frame. This criterion is particularly effective at suppressing false-positive responses on the venue's rear-wall graphics, which are static and therefore consistently absent from the foreground mask.

The overall candidate score combines YOLO confidence, foreground support, motion consistency with the recent trajectory, Kalman prediction error, and a penalty for candidates that have remained stationary across multiple frames. The candidate with the highest composite score is selected as the accepted ball position for that frame, subject to passing threshold checks. Confidence thresholds are adaptive: a lower threshold is used during recovery periods when the tracker has recently lost the ball, and a higher threshold is required during stable tracking.

4.6.3 Speed-Based Outlier Rejection

A complementary validation criterion enforces physical plausibility. The displacement between a candidate detection and the last accepted ball position is computed in pixels per frame and compared against a maximum plausible speed derived from the known upper bound on volleyball speed and the camera frame rate. Candidates that imply a displacement exceeding this bound are rejected as outliers regardless of their YOLO confidence score (Kamble et al., 2019). This filter is particularly effective at eliminating isolated false-positive detections that appear in spatially unrelated regions of the frame with no causal connection to the recent trajectory.

4.6.4 Gap Handling and Recovery

Short detection gaps — frames in which no candidate passes the acceptance criteria — are handled without terminating the tracked trajectory. During a gap, the tracker continues to advance the Kalman prediction and retains the most recent accepted positions as reference points. If the gap is short enough, the missing trajectory segment is infilled by interpolation between the last accepted position before the gap and the first accepted position after it, producing a smooth trajectory without artificial breaks.

When a gap extends beyond the infill threshold, the tracker enters a recovery state in which the confidence requirements for acceptance are relaxed and the search region around the predicted position is widened. This behaviour is important for volleyball footage because the ball is frequently occluded by players during net exchanges, which are precisely the frames most relevant to event classification. The maximum gap length before the trajectory is considered lost is configurable, and a hard physics-based rejection gate (`GAME_RULES_PHYSICS_REJECT_ENABLED`) is available but disabled by default.

Figure 4.2 shows an example of the tracking preview generated by the implementation, including accepted ball detections and the reconstructed trajectory over a rally.

Figure 4.2: Example of the ball tracking preview generated by the system, showing accepted ball positions and the reconstructed recent trajectory over the analysed rally.

4.7 Block Detection Implementation

Block detection is implemented as a dedicated module (`block_detection.py`) that operates in parallel with the ball tracker and provides an independent assessment of whether a blocking action occurred during the current rally. It receives ball positions and court geometry information on each frame and internally maintains a state machine that progresses through stages including detection of an attacking approach, identification of net contact or near-net trajectory interruption, and confirmation of a block outcome.

The module exposes two interfaces to the analytics engine: a live assessment that is updated on each frame during an active rally, and a final assessment that is produced when the rally is closed. The final assessment is passed to the post-mortem event classifier and may promote the rally's event category to `block` if the geometric evidence is sufficiently strong. Because the module operates on trajectory evidence rather than direct visual recognition of players' arms or hands, its reliability is coupled to the quality of the ball trajectory in the frames surrounding the net contact.

4.8 Scoreboard Reading Implementation

4.8.1 Scoreboard ROI Selection

Before the analysis loop begins, the operator selects the scoreboard region of interest by drawing a bounding rectangle around the scoreboard area in the first processed frame, using `cv2.selectROI`. The selected rectangle is automatically refined by `refine_scoreboard_roi`, which adjusts the boundaries based on local image content to achieve a tighter fit around the actual scoreboard digits. The refined region is then subdivided into four subregions corresponding to the sets score and points score for each team: `sets_a`, `points_a`, `sets_b`, and `points_b`.

4.8.2 Digit Template Matching

Digit recognition is performed by template matching against a library of ten reference images (`0.png` through `9.png`) stored in the `digit_templates` directory. These templates were prepared from the scoreboard style used during development at *Pav. Arq. Jerónimo Reis Espinho* and capture its specific font, size, and rendering style. At runtime, each digit sub-region is preprocessed — including resizing and binarisation — and then matched against the template library using `cv2.matchTemplate`. The template with the highest normalised cross-correlation score is selected as the recognised digit for that sub-region (Guo et al., 2011).

This approach is deliberately domain-specific. By using templates derived from the actual scoreboard rather than a generic digit font, the matcher achieves stable recognition under visual conditions similar to the development footage without requiring a general-purpose scene text recogniser.

4.8.3 Temporal Voting

Because individual frame reads are noisy, the scoreboard reader does not accept a raw digit reading as the current score immediately (Guo et al., 2011). Instead, readings are accumulated in a sliding vote window. A candidate score value is confirmed only when it accumulates a sufficient number of consistent votes within the window. The voting parameters — window size, minimum hit count, and the number of confirmations required for an initial read and for a score-change read — are configured in `config.py`. This mechanism suppresses the transient misreadings caused by compression artefacts, brief illumination changes, or minor camera vibration, without requiring every frame to be read.

The scoreboard is sampled periodically rather than on every frame; the sampling interval is controlled by the `ocr_every_n_frames` parameter. This reduces the computational overhead of the OCR stage while maintaining sufficient temporal resolution to detect score changes reliably.

Figure 4.3 illustrates the selected scoreboard region and examples of the digit templates used by the template-matching reader.

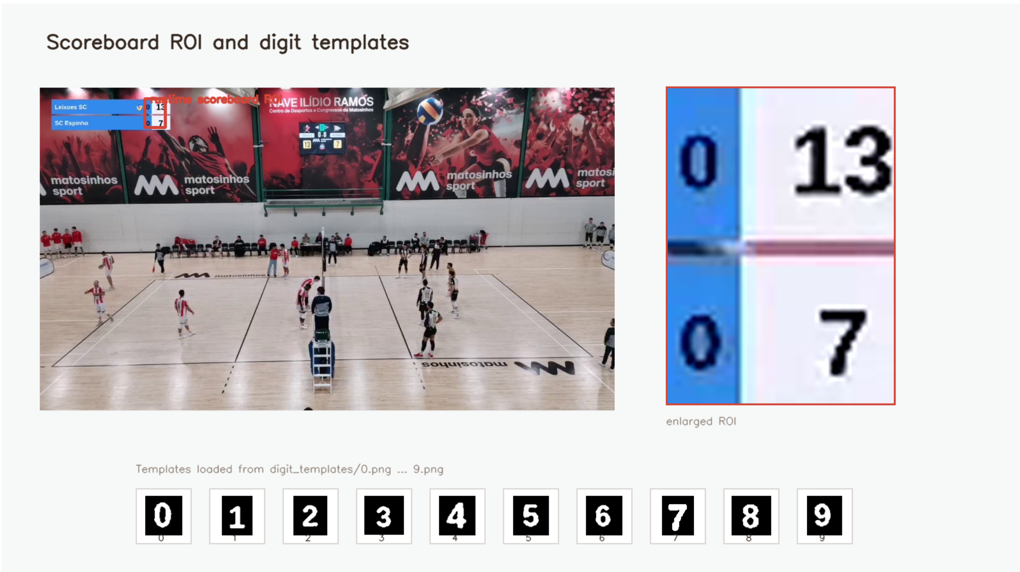


Figure 4.3: Example of the selected scoreboard region of interest and the reference digit templates used by the template-matching reader to obtain temporally validated score values.

4.8.4 Score Progress Validation

A confirmed vote is not automatically accepted as a rally boundary. An additional logical validation step checks that the candidate score represents a plausible progression from the last confirmed score. Specifically, a new score is accepted only if exactly one team's score has increased by one point and the other team's score is unchanged. Readings that imply a score decrease, a jump of more than one point, or a simultaneous change for both teams are rejected as noise. This constraint leverages the rules of volleyball scoring to filter out residual misreadings that survive the voting stage (Guo et al., 2011).

When a logically valid $+1$ transition is confirmed, the analytics engine treats it as a rally boundary, finalises the current rally, and triggers event classification for that rally.

4.9 Rally Management and Event Classification

4.9.1 Rally Boundary Detection

Rally lifecycle management is handled by the `AnalyticsEngine` in `analytics.py`. A rally is opened when the system detects the start of play and `RallyManager.start_if_needed()` is called. It remains open while the ball is in play and while the scoreboard has not yet confirmed a score change. When the scoreboard module reports a validated $+1$ transition, the current rally is finalised: the accumulated trajectory data is frozen, the winning team is determined from the score change, and the post-mortem classification routine is invoked.

4.9.2 Court-Geometry-Based Reasoning

Post-mortem event classification is performed by `check_post_mortem()`, which analyses the frozen rally trajectory in court coordinates. The analysis examines the ball's path relative to the net plane, the court boundaries, and the court sides. Key questions include: did the ball cross the net, and in which direction; was the ball's final position inside or outside the court; how close to the net was the ball's last detected position before the rally ended; and which team was in possession before the point was awarded? The block detector's final assessment is also consulted at this stage. All spatial reasoning is performed in the metric court coordinate system established during calibration, not in raw image pixels.

4.9.3 Event Taxonomy and Rules

The classification rules map the geometric evidence and possession context to a point type. The primary categories assigned to rally records are:

- **`attack`**: includes successful attacking actions and, in the evaluated output mapping, also absorbs spike-like and freeball-like outcomes when they result in an offensive rally-ending action.
- **`block`**: confirmed by the block detection module based on net-contact trajectory evidence during the rally.

- **`ball_out`**: the ball's last detected position was outside the court boundaries without a valid net crossing.
- **`ball_on_net`**: the trajectory terminated at or near the net plane with no crossing detected.
- **`ace`**: reserved for serve-related points when the available evidence supports this classification.
- **`error`**: a rally termination that does not match the geometric signatures of the above categories.

The codebase also includes additional internal labels, including `ace`, `freeball`, `spike`, and `undefined`. In the evaluated version of the prototype, these labels are consolidated where appropriate. Spike-like and freeball-like outcomes are reported under the broader `attack` category, reducing fragmentation in the output statistics and making the evaluation more consistent given the limited number of reviewed rallies. The `undefined` label is reserved for internal fallback cases and is not treated as a final event category in the reported evaluation. Each classified event is stored with a `confidence` value, a `reason` string, and a `notes` field that traces the evidence used in the classification decision, supporting later inspection and debugging.

4.9.4 Rule-Based Classification Procedure

Figure 4.4 summarises the order in which the rule-based classifier evaluates the available evidence after a rally has been closed by a validated scoreboard change.

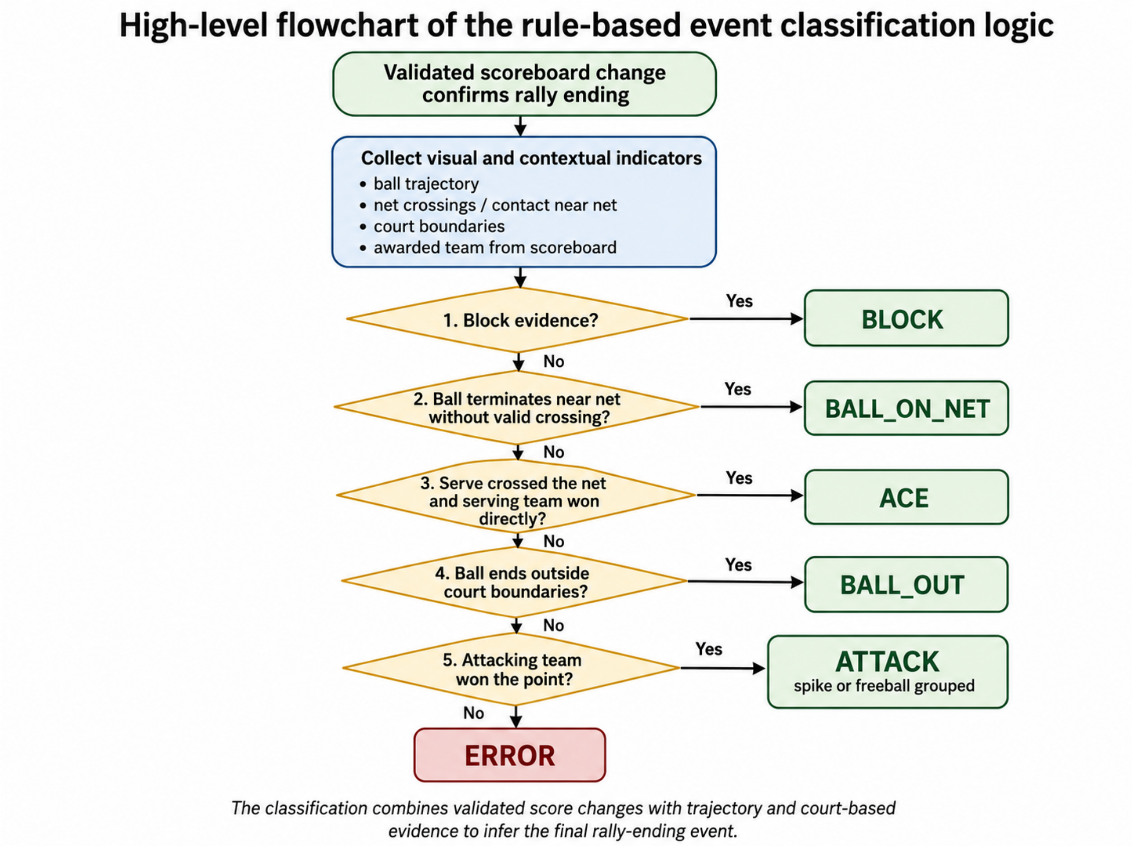


Figure 4.4: High-level flowchart of the rule-based rally-ending event classification logic. The classification is triggered by a validated scoreboard change and combines ball trajectory, net-related indicators, court boundaries, and the awarded team to infer the final event category.

Classification is therefore not attempted solely from an isolated frame or an unconfirmed interruption in play. A validated score transition first confirms the rally ending and determines the team awarded the point. The classifier then collects visual and contextual indicators from the final rally segment, including the ball trajectory, detected net crossings, possible contact close to the net, the final position relative to the court boundaries, possession context, and the awarded team. Conditions supported by stronger geometric evidence — notably `block`, `ball_on_net`, and `ball_out` — are evaluated before broader attacking or fallback categories. The serve context is also considered when the available evidence supports an `ace`.

When the trajectory evidence alone is insufficient, the awarded team provides an auxiliary decision criterion. Spike-like and freeball-like attacking outcomes are consolidated into the final `attack` category. If the attacking team wins the point, this context supports an `attack` decision; if the opponent wins and no more specific geometric condition is satisfied, the rally can be assigned to `error`. This ordering favours specific, interpretable evidence while retaining a deterministic fallback for incomplete trajectories.

Algorithm — Rally-ending event classification. **Input:** ball trajectory, court geometry, net position, validated scoreboard change, and awarded team.

Output: rally-ending event category.

1. Confirm that a valid scoreboard change has occurred; otherwise, defer classification until the rally boundary is validated.
2. Collect trajectory, net-crossing, net-contact, court-boundary, possession, and awarded-team indicators from the final rally segment.
3. If the block detector and near-net trajectory provide block evidence, classify the rally as `BLOCK`.
4. Otherwise, if the ball terminates near the net without a valid crossing, classify the rally as `BALL_ON_NET`.
5. Otherwise, if the serve crossed the net, no valid defensive return is detected, and the serving team won directly, classify the rally as `ACE`.
6. Otherwise, if the final trajectory ends outside the calibrated court boundaries, classify the rally as `BALL_OUT`.
7. Otherwise, if the attacking team won the point, classify spike-like or freeball-like evidence under the final category `ATTACK`.
8. Otherwise, classify the rally as `ERROR`.

These rules form an interpretable approximation of rally-ending events; they are not intended to model every tactical detail or intermediate action in volleyball. Table 4.2 summarises the principal evidence and auxiliary criterion associated with each reported category.

Table 4.2: Summary of the principal rules used for rally-ending event classification.

Event category	Main evidence used	Auxiliary criterion
`block`	Ball interaction or trajectory pattern close to the net consistent with a block.	Awarded team and final rally context.
`ball_on_net`	Ball terminates near the net without a valid crossing.	Absence of a valid continuation after net contact.
`ace`	Serve crosses the net and no valid defensive return is detected.	Serving team wins the point.
`ball_out`	Final ball trajectory indicates that the ball leaves the calibrated court boundaries.	Point awarded to the opponent of the last attacking action.
`attack`	Valid attacking action after a net crossing, including spike-like or freeball-like outcomes.	Attacking team wins the point.
`error`	No stronger event condition is satisfied by the available evidence.	Opponent wins the point or the visual evidence is insufficient.

4.10 Output Generation

4.10.1 CSV Rally Summary

At the end of the analysis run, the rally manager serialises the collection of finalised rallies to `outputs/tese_volleyball_stats.csv` using Pandas. Each row in the file corresponds to one confirmed rally and records the rally identifier, start and end timestamps, total duration, rally duration, the identified attacker, the point type, the winning team, the number of net crossings, and ball speed statistics (mean and maximum) for that rally. The file is written once at the end of the run and reflects all rallies finalised during the processed segment.

4.10.2 JSON Event Store

Each confirmed rally event is recorded in `outputs/volleyball_events.json` by `EventStore.record_-event()`. The file is updated atomically on each event, so it reflects the current state of the run at any point during processing. Because `event_store_reset_on_start` is `True` by default, the store is cleared at the beginning of each run, ensuring that results from different analysis sessions do not accumulate in the same file.

Each event record contains a rich set of fields: event identifier, type, category, point type, start and end frame indices, timestamps, rally duration, ball speed statistics, team and side assignments, confidence, reason, notes, and the path to the associated preview image. This structured record is the primary machine-readable output of the pipeline and supports detailed per-event inspection independently of the rally summary.

4.10.3 Event Preview Images

For each recorded event, the system saves an annotated frame to `outputs/event_previews/`, named sequentially (`event_0001.png`, `event_0002.png`, and so on). The preview image captures the field state at the moment the event was classified, with visual overlays indicating the ball position, court geometry, and relevant trajectory information. The path to the preview image is stored in the corresponding JSON event record under `representative_frame_path`.

4.10.4 Evaluation Summary Output

When the pipeline is invoked with the `--eval` flag, an additional summary file `outputs/stats_summary.json` is written at the end of the run. This file contains aggregate counts of the main event categories, per-team point totals, and a snapshot of the final score. It is intended as a convenience output for development and Thesis evaluation purposes and is not produced during a standard analysis run.

4.11 Implementation Limitations

Several implementation constraints are acknowledged here as engineering trade-offs rather than fundamental design failures.

The pipeline requires manual initialisation at the start of each run. Court calibration and scoreboard region selection cannot be automated without additional infrastructure, such as automatic court line detection or scoreboard localisation, which are not part of the current implementation.

The ball detector and the digit templates are tailored to the development footage from *Pav. Arq. Jerónimo Reis Espinho*. However, the screenshots and walkthrough examples shown in this Chapter were generated from the active footage at the time of documentation, recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. Applying the system to another venue or a different broadcast overlay would require checking the calibration, retraining or fine-tuning the detector, and replacing the template library where necessary.

The hard physics-based ball rejection gate (`GAME_RULES_PHYSICS_REJECT_ENABLED`) is disabled by default. Although it would provide an additional layer of plausibility filtering, it was not enabled in the active configuration at the time of writing.

An exploratory Detectron2-based detection pipeline was also implemented in a separate experimental directory. However, this branch was not integrated into the active runtime pipeline and was only used for a preliminary qualitative assessment. Based on this initial exploration, YOLOv8 was retained as the main detection backend because it was already better integrated into the prototype and more suitable for the subsequent tracking and event-classification stages. A rigorous comparison between YOLOv8 and Detectron2 is therefore left for future work.

Finally, no quantitative benchmark against an independently annotated ground truth has been performed at the time of writing. The evaluation presented in Chapter 5 is based on manual review of two sets recorded at *Pav. Arq. Jerónimo Reis Espinho* and *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*.

This Chapter has described the realisation in code of the system specification defined in Chapter 3. The implementation follows a modular sequential pipeline in which court calibration, ball detection, Kalman- assisted tracking, foreground-based validation, template-based scoreboard reading, rally management, heuristic event classification, and persistent output generation are each handled by a dedicated component. Manual initialisation is required at startup, but the analysis stage operates automatically once the operator has completed the calibration and region-selection steps.

The following Chapter presents the evaluation of the implemented system on two sets from different venues and discusses the observed behaviour of the detection, tracking, scoreboard reading, and event classification components.

Chapter 5

Evaluation

The evaluation applies the implemented system to recorded footage from two competitive volleyball matches. The methodology, test scenarios, results, and observed error patterns are described in turn, followed by a discussion of the system's behaviour and the limitations inherent to this evaluation.

The evaluation reported in this Chapter focuses on the active YOLOv8-based pipeline. The exploratory Detectron2 branch was not included in the final evaluation.

5.1 Evaluation Methodology

The evaluation was conducted by direct observation of two recorded match segments together with the system's classification output during processing. For each point scored within the observed segment, the reviewer recorded the resulting score, the type of action that ended the rally, and the team awarded the point, based on the recorded footage. This observation was then compared, point by point, against the classification produced by the system for the corresponding rally.

A point was marked *correct* when the reviewer's judgement of the rally-ending event agreed with the system's event classification, and *incorrect* otherwise. In this evaluation, "incorrect" refers primarily to disagreement in the event category assigned to a rally, such as classifying an attack as a block or confusing `ball_out` with `ball_on_net`. The winning team was inferred from the validated scoreboard change and was therefore generally known to the system once a rally boundary was confirmed. Accordingly, the disagreement counts reported in the tables should be interpreted mainly as event-type classification errors, not as point-award errors. Cases in which the system produced no corresponding classification for a reviewed point are discussed separately when they occurred. Where the likely cause of a disagreement could be identified from the footage, a short explanatory note was recorded (for example, indicating that the rally in question was unusually fast, or that additional balls were visible near the court).

This evaluation is end-to-end: it reflects the correctness of the final classified output produced during processing, rather than isolating the contribution of individual pipeline stages (ball detection, tracking, scoreboard reading, or event classification). The event categories referred to in this Chapter follow the taxonomy described in Chapter 4: `attack`, `block`, `ace`, `ball_out`, `ball_on_net`, and `error`. For evaluation purposes, the reported categories correspond to the final output mapping of the prototype. Internal labels such as `spike` and `freeball` are grouped under `attack`, while `undefined` is treated as an internal fallback rather than a reported event category.

5.2 Test Scenarios

Two evaluation scenarios were used, corresponding to one set from two different matches recorded at two different venues.

Match 1 corresponds to a set played between AA Espinho / Académica and Clube Kairós, recorded at *Pav. Arq. Jerónimo Reis Espinho*. The configured analysis window for this footage spans from 00:26:30 to 00:47:17, corresponding to approximately twenty minutes and forty-seven seconds (37,410 frames at 30 frames per second).

Match 2 corresponds to a set played between Espinho and Leixões, recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. The main practical difference from Match 1 was the visual background of the venue: the wall behind the court contained more complex visual patterns and background clutter. The configured analysis window for this footage spans from 00:26:00 to 00:46:27, corresponding to approximately twenty minutes and twenty-seven seconds (36,810 frames at 30 frames per second).

Cross-Venue Reconfiguration

Both evaluated sets were processed with the same overall pipeline and a similar fixed-camera recording setup. The court and net calibration procedure described in Section 4.4 is performed interactively at the start of every run: the operator selects four court corners and two top-of-net points, from which a homography is computed and persisted, overwriting the calibration file from any previous run. For Match 1, calibration was performed for *Pav. Arq. Jerónimo Reis Espinho*; for Match 2, the procedure was repeated for *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*, producing a calibration specific to that venue. The scoreboard region of interest is likewise selected interactively at the start of every run, as described in Section 4.8.1.

The main practical difference between the two scenarios was the visual background of the venue. Match 1 was recorded at *Pav. Arq. Jerónimo Reis Espinho*, whose background was visually simpler and more closely represented in the dataset and tuning process. Match 2 was recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*, where the wall behind the court contained more complex visual patterns and background clutter. This increased the likelihood of false positives or ambiguous trajectory evidence during some rallies.

5.3 Results

The disagreement counts in Tables 5.1 and 5.2 refer to event-category disagreements under the manual review protocol, rather than to systematic errors in assigning points to teams.

5.3.1 Match 1: AA Espinho vs. Clube Kairós

Table 5.1 summarises the results obtained for Match 1, broken down by event category. Of the 44 points reviewed, 36 were classified in agreement with the reviewer's observation, corresponding to an overall manual-review agreement rate of 81.82%.

Table 5.1: Match 1 per-category agreement.

Event category	Total	Disagreements	Disagreement rate
`attack`	23	3	13.04%
`ball_out`	16	5	31.25%
`block`	2	0	0.00%
`ball_on_net`	3	0	0.00%
Overall	**44**	**8**	**18.18%**

5.3.2 Match 2: Espinho vs. Leixões

Table 5.2 summarises the results obtained for Match 2, broken down by event category. Of the 40 points reviewed, 29 were classified in agreement with the reviewer's observation, corresponding to an overall manual-review agreement rate of 72.50%.

Table 5.2: Match 2 per-category agreement.

Event category	Total	Disagreements	Disagreement rate
`attack`	24	5	20.83%
`ball_out`	9	3	33.33%
`block`	3	1	33.33%
`ball_on_net`	2	1	50.00%
`ace`	1	0	0.00%
`error`	1	1	100.00%
Overall	**40**	**11**	**27.50%**

5.3.3 Overall Agreement

Across the two evaluated sets, the system's classification agreed with the reviewer's observation for 65 of 84 points (77.38%). Agreement was higher for Match 1 (81.82%) than for Match 2 (72.50%), a difference that is discussed further in Section 5.5.

Table 5.3: Overall agreement.

Evaluation set	Reviewed points	Agreements	Agreement rate
Match 1	44	36	81.82%
Match 2	40	29	72.50%
Combined	**84**	**65**	**77.38%**

Figure 5.1 provides a visual comparison of the manual-review agreement rates obtained for the two evaluated match segments and for the combined result.

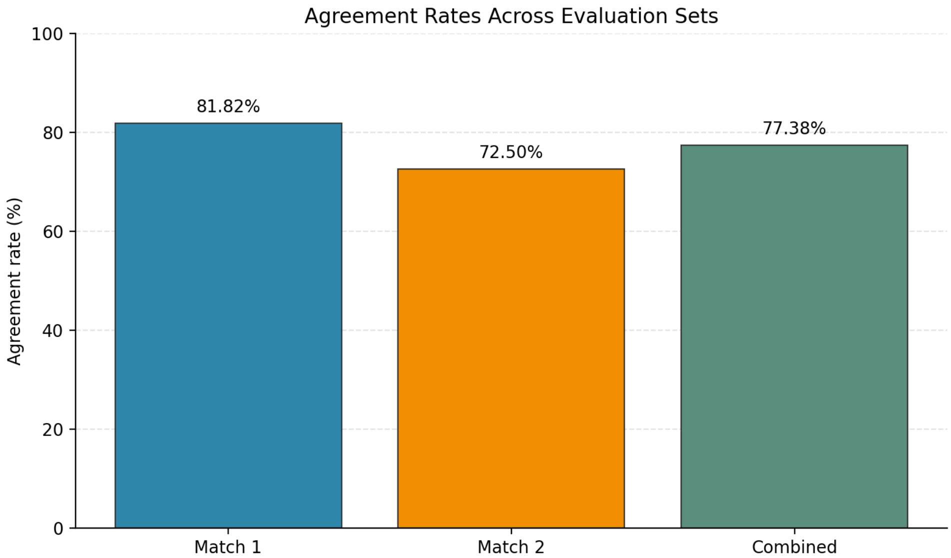


Figure 5.1: Comparison of the manual-review agreement rates obtained for Match 1, Match 2, and the combined set of 84 reviewed points.

5.4 Error Analysis

The explanatory notes recorded during the manual review point to a small number of recurring patterns. These patterns are described qualitatively below; as noted in Section 5.1, "incorrect" is an informal, holistic label rather than a member of a closed error taxonomy, so the categorisation below should be read as descriptive rather than exhaustive. The error patterns discussed below therefore concern the interpretation of the rally-ending action, while point attribution itself was driven by the validated scoreboard transition.

Confusion near the net. The `ball_out` category shows the highest disagreement rate in both matches (31.25% and 33.33%), and the single `ball_on_net` disagreement in Match 2 was itself a confusion with `ball_out`. This is consistent with the discussion in Chapter 4: both categories are determined by the final position of the ball trajectory relative to the net plane and the court boundary in normalised court coordinates, and the geometric margins separating the two outcomes are narrow. Small errors in the tracked trajectory near the end of a rally are therefore more likely to produce a confusion between these two categories than an unrelated classification.

Fast rallies. Several disagreements in both matches were annotated as occurring during unusually fast exchanges. This is consistent with the small-object detection and motion-blur challenges discussed in Chapter 3: at high ball speeds, the detector is more likely to miss frames or produce noisy candidates, leaving the tracker's gap-recovery mechanism (Section 4.6.4) with less trajectory evidence at the point of net contact or boundary crossing.

Visual interference from extraneous objects. One disagreement in Match 1 was attributed to additional balls being visible near the court during the rally, likely corresponding to warm-up balls or balls retrieved by ball collectors. Because these objects are themselves volleyballs, they are not distinguishable from the ball in play on the basis of appearance alone, and are not addressed by the venue-specific background images used during training of the ball detector.

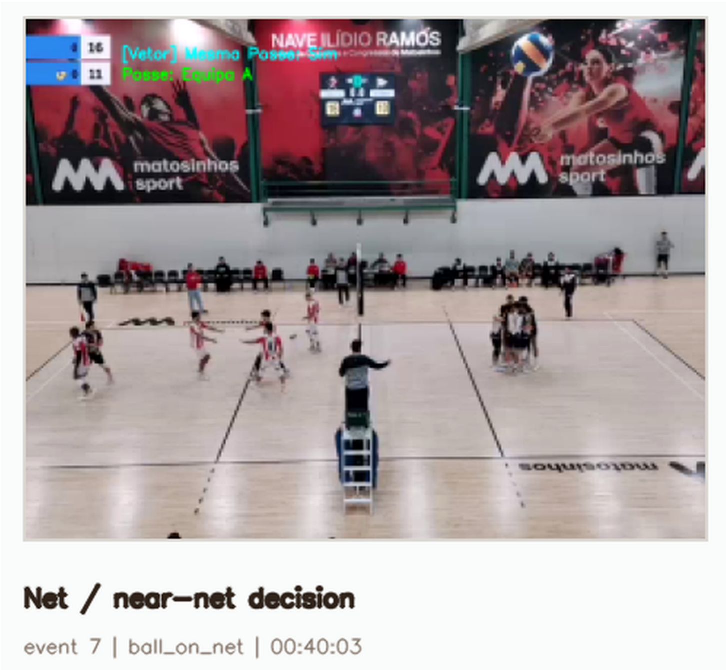
Missed classification. One point in Match 2, annotated as an `error`-type rally ending, was not associated with any classification by the system. This represents a case in which the system produced no output corresponding to the reviewed point, rather than a confusion between two defined categories.

Block category. The `block` category shows a markedly different disagreement rate between the two matches (0.00% in Match 1 versus 33.33% in Match 2). Given that only two and three block events, respectively, were observed in the two sets, these figures should be interpreted with caution: the sample size is too small to support a reliable estimate of block classification accuracy.

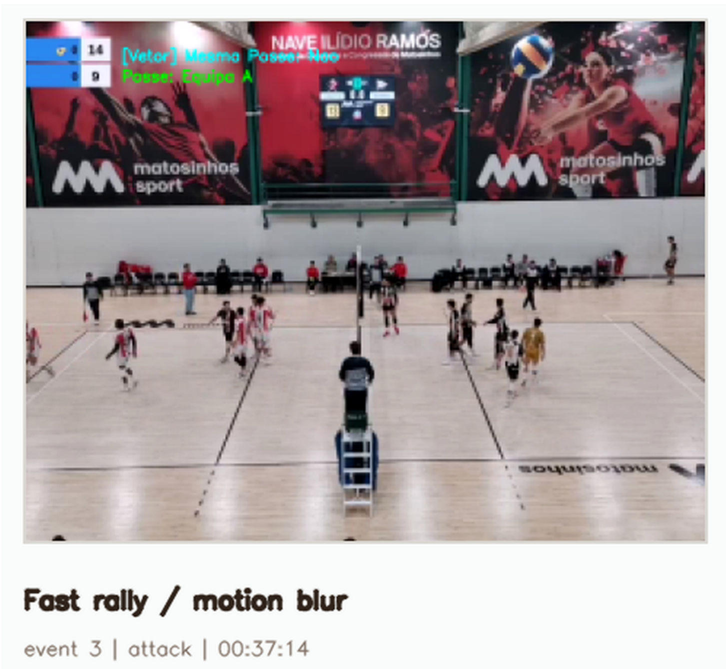
Table 5.4: Qualitative error patterns.

Pattern	Interpretation
Confusion near the net	Boundary and net-plane decisions are sensitive to small trajectory errors at the end of rallies.
Fast rallies	Motion blur and missed detections reduce the evidence available to the tracker and classifier.
Extraneous balls	Additional volleyballs near the court can be visually indistinguishable from the ball in play.
Missed classification	Some reviewed points may not receive a corresponding system output.
Small block sample	Block accuracy estimates are unstable because very few block events were observed.

Figure 5.2 presents representative examples of the main error patterns identified during manual review.



(a) Near-net trajectory ambiguity affecting the distinction between net contact and an out-of-court ending.



(b) Fast rally in which motion blur and sparse detections reduce the trajectory evidence available to the classifier.

Figure 5.2: Representative error cases identified during manual review, illustrating near-net ambiguity and reduced trajectory evidence during fast play.

(c) Visual interference from an additional ball-like object that may compete with the ball in play during detection and tracking.

Figure 5.2: Representative error cases identified during manual review (continued), showing interference from an additional ball-like object near the court.

5.5 Discussion

The overall agreement figures — 81.82% for Match 1 and 72.50% for Match 2 — indicate that the heuristic, geometry-based event classification approach described in Chapter 4 agrees with manual observation for a majority of points on both evaluated sets, while leaving a substantial fraction of points in disagreement.

This result should be interpreted as a preliminary end-to-end validation rather than as a complete statistical characterisation of the system. The evaluation measures the behaviour of the active pipeline as a whole, from ball detection and tracking through scoreboard validation and event classification, and therefore does not identify the isolated contribution of each component.

The categories `attack` and `ball_out` dominate the reviewed sample in both matches, so their disagreement rates provide the most stable indication of the system's behaviour in this evaluation. By contrast, categories such as `block`, `ace`, and `error` occur only a small number of times; their percentages are useful for identifying possible failure modes but should not be interpreted as reliable category-level performance estimates.

The lower agreement rate for Match 2 is notable in light of the change of venue. The same overall pipeline and recording approach were used, but the visual background differed between the two pavilions. Match 1 was closer to the development and tuning conditions: the background behind the court was simpler and better represented in the custom dataset. By contrast, Match 2 was recorded in a venue with more visually complex wall patterns and background clutter. This likely made ball detection and trajectory interpretation more difficult in some rallies, particularly when combined with fast ball motion or ambiguous positions near the net and court boundaries. This explanation should be interpreted as a plausible observation from the evaluated footage rather than as an isolated component-level measurement.

More broadly, the fact that the system could be applied to a second venue at all — by repeating the court calibration procedure while reusing the existing detection model and runtime pipeline — indicates that the pipeline's modular architecture permits some degree of adaptation to a different venue without modification to the underlying detection, tracking, or classification logic. This should not be read as evidence of automatic generalisation: the calibration step still required the same manual effort as for the first venue, and the results for Match 2 suggest that visually complex venue backgrounds may require additional dataset coverage and tuning.

The difference between the two matches also reinforces the sensitivity of the pipeline to changes in venue-specific background appearance. Although the results do not replace a component-level evaluation, they show that the integrated pipeline can produce useful rally-level classifications for a substantial portion of the reviewed points.

The concentration of disagreements around the `ball_out` / `ball_on_net` boundary, and around fast rallies more generally, points towards the geometric margin used to distinguish these categories and the trajectory recovery behaviour during high-speed play as promising targets for future refinement of the event classification logic.

5.6 Limitations of the Evaluation

This evaluation is subject to several limitations that should be considered when interpreting the results above.

Sample size. The evaluation covers a single set from each of two matches, totalling 84 points. This sample size is small, and the per-category breakdowns — in particular for `block`, `ace`, and `error`, each represented by one to three observations — do not support statistically robust conclusions. The evaluation should be interpreted as a preliminary end-to-end assessment rather than a complete statistical validation of the system. Only two sets were reviewed, and several event categories occurred only a small number of times. As a result, the reported agreement rates are useful for understanding the behaviour of the prototype, but they are not sufficient to establish general performance across teams, venues, or match conditions.

Single-reviewer judgement. The reference observations used for comparison were recorded by a single reviewer, without an independent second annotator or a formal inter-rater agreement check. The classification of a point as "correct" or "incorrect", and the explanatory notes attached to disagreements, therefore reflect a single observer's judgement, and "incorrect" was used as an informal label rather than a member of a closed error taxonomy, as discussed in Section 5.1.

End-to-end evaluation. The evaluation does not isolate the performance of individual pipeline components. A disagreement between the system's classification and the reviewer's observation may originate from the ball detector, the tracker, the scoreboard reader, the court calibration, or the event classification rules, individually or in combination. Component-level metrics — such as detection precision and recall, tracking continuity, or scoreboard reading accuracy in isolation — were not computed as part of this evaluation.

No external benchmark. No comparison against an existing commercial or research system was performed. The agreement figures reported here describe the behaviour of the implemented pipeline on the two evaluated sets and should not be interpreted as a benchmark against alternative approaches.

This Chapter has presented an evaluation of the implemented system on two recorded sets from different matches and venues. The system's classification agreed with manual observation for 81.82% of points in the set recorded at *Pav. Arq. Jerónimo Reis Espinho* and for 72.50% of points in the set recorded at *CDC Matosinhos – Nave Ilídio Ramos Matosinhos*. Error analysis identified the boundary between `ball_out` and `ball_on_net`, fast rallies, and visual interference from extraneous balls as recurring sources of disagreement. The system's application to a second venue required repeating the court and net calibration procedure but reused the existing detection model and runtime pipeline without modification; the lower agreement rate observed for the second venue suggests that visually complex background conditions may require additional dataset coverage and tuning.

The final Chapter of this dissertation summarises the contributions of this work, reflects on the limitations identified throughout, and outlines directions for future development.

Chapter 6

Conclusion

The conclusion summarises the work developed, reflects on the main results and limitations, and identifies directions for future extension. The focus is on the practical lessons learned from building and evaluating a semi-automatic volleyball video analysis pipeline under realistic single-camera indoor conditions.

6.1 Main Findings

This dissertation set out to investigate whether useful volleyball performance statistics could be extracted semi-automatically from standard recorded match footage. The implemented system demonstrates that a modular computer vision pipeline can combine ball detection, temporal tracking, scoreboard reading, manual court calibration, and rule-based event classification into a coherent workflow that produces structured CSV and JSON outputs.

The evaluation showed that the system agreed with manual observation for a majority of reviewed points across two match segments. The results were stronger on the first evaluated venue than on the second venue, which supports the central assumption of the work: venue-specific visual conditions, especially the complexity of the background behind the court and its similarity to the footage used during dataset construction and parameter tuning, have a significant influence on reliability. The system is therefore best understood as a semi-automatic research prototype that can support post-match analysis, rather than as a fully general commercial tool.

6.2 Contributions

The work contributes an end-to-end volleyball analysis pipeline tailored to real indoor footage. Unlike a single isolated model, the proposed system integrates several complementary components: a fine-tuned YOLOv8 detector for the ball, Kalman-assisted tracking, foreground validation, scoreboard digit reading with temporal voting, homographic court calibration, and geometry-based event inference.

A second contribution is the emphasis on practical deployment constraints. The system explicitly handles a user-selected processing window, stores calibration parameters, writes machine-readable outputs, and generates preview artefacts that can be inspected after execution. These design choices make the pipeline more usable in a club-level analysis workflow, where complete automation is less important than producing structured information with limited manual effort.

A third contribution is the documentation of limitations encountered during the implementation and evaluation. False positives and ambiguous trajectory evidence caused by visually complex venue backgrounds, motion blur during fast rallies, and narrow geometric margins near the net were all identified as important failure modes. These observations provide concrete guidance for future work.

6.3 Limitations

The system remains limited in several respects. First, it requires manual setup: the court geometry and scoreboard region must be selected interactively before analysis. Second, the detector remains sensitive to the visual conditions of the footage used during development, especially the complexity of the background behind the court. Applying the pipeline to a new venue may require additional annotation, retraining, or targeted background samples; scoreboard template preparation may also be needed when the scoreboard format changes.

Third, the event classifier is heuristic and geometry-based. This makes its behaviour interpretable, but also means that ambiguous cases near the net or near the court boundary are sensitive to tracking noise and calibration errors. Finally, the evaluation was limited to two match segments and did not isolate the performance of each pipeline component independently.

6.4 Future Work

Future work should first address the robustness of ball tracking under more diverse venue backgrounds. Future work should evaluate the system on more venues with different background complexity levels and should expand the training dataset with additional negative/background samples from visually complex pavilions. Improvements to scoreboard reading remain relevant, especially for venues with different scoreboard formats, but the evaluated difference between the two matches mainly points to background complexity.

Future work should also include a broader evaluation protocol covering more matches and venues with different background complexity levels. A larger manually annotated ground truth would make it possible to report more stable per-category results and to evaluate individual components such as ball detection, tracking, scoreboard reading, and event classification separately. This would also support a more rigorous comparison between YOLOv8, Detectron2, and other detection backends under the same dataset and evaluation conditions. Although an exploratory Detectron2-based branch was implemented during development, it was not fully integrated into the final runtime pipeline or evaluated under the same conditions as the YOLOv8-based detector.

A third direction is to improve visual and operational usability. Automatic court line detection, automatic scoreboard localisation, richer event preview images, and an interactive review interface would reduce manual setup effort and make the system easier for coaches or analysts to use in practice.

Taken together, the work shows that semi-automatic volleyball analysis from single-camera footage is feasible, but that reliable deployment depends on careful integration of detection, tracking, scoreboard validation, calibration, and domain-specific event logic. It therefore provides a foundation for future systems that can reduce manual annotation effort while preserving the interpretability required in sports performance analysis.

Bibliography

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. `10.1109/ICIP.2016.7533003`.

Chen-Ni Chen and Wei-Ta Chu. How it flies and why it flies? volleyball trajectory segmentation and classification. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(5):1591–1595, 2021. `10.1109/TCSII.2021.3067027`.

Xina Cheng, Linzi Liang, and Takeshi Ikenaga. Automatic data volley: Game data acquisition with temporal-spatial filters. *Complex & Intelligent Systems*, 8(6):4993–5010, 2022. `10.1007/s40747-022-00752-3`.

Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9921–9931, 2023. `10.1109/ICCV51070.2023.00910`.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. `10.1109/CVPR.2005.177`.

Keisuke Fujii. Computer vision for sports analytics. In *Machine Learning in Sports*, SpringerBriefs in Computer Science, pages 21–57. Springer, 2025. `10.1007/978-981-96-1445-5_2`.

Fédération Internationale de Volleyball. Volleyball Information System. `https://www.fivb.org/en/volleyball/VIS.asp`, 2026. Accessed: 2026-06-22.

Silvio Giancola, Anthony Kang, Jiaming Feng, Anthony Cioppa, Marco Bertini, Ivo Danihelka, Ali Diba, Sergio Fernández, Bernard Ghanem, Irtiza Hasan, et al. DeepSportradar-v1: Computer vision dataset for sport understanding with high resolution images. In *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, pages 1–8, 2022. `10.1145/3552437.3555699`.

Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 10.1109/ICCV.2015.169.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 10.1109/CVPR.2014.81.

Jinlin Guo, Cathal Gurrin, Songyang Lao, Colum Foley, and Alan F. Smeaton. Localization and recognition of the scoreboard in sports video based on SIFT point matching. In *Advances in Multimedia Modeling*, volume 6524 of *Lecture Notes in Computer Science*, pages 337–347. Springer, 2011. 10.1007/978-3-642-17829-0_32.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, et al. Array programming with NumPy. *Nature*, 585:357–362, 2020. 10.1038/s41586-020-2649-2.

Guanghui Huang. An effective volleyball trajectory estimation and analysis method with embedded graph convolution. *International Journal of Distributed Systems and Technologies*, 14(2):1–13, 2023. 10.4018/IJDST.317936.

Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsi-Ui Ik, and Wen-Chih Peng. TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2019. 10.1109/AVSS.2019.8909871.

Mostafa S. Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *Computer Vision – ECCV 2018*, pages 742–758, 2018. 10.1007/978-3-030-01219-9_44.

Mostafa S. Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1971–1980, 2016. 10.1109/CVPR.2016.217.

Mehmet Görkem .Işgüzar, Ifet Mahmutović, Serdar Uslu, and Rıza Barak. Relationship between stoppage time, rally length, and selected performance variables in elite men's volleyball competitions. *Annales Kinesiologiae*, 16(2):117–129, 2026. 10.35469/ak.2025.545.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2024.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 10.1115/1.3662552.

Prashant R. Kamble, Avinash G. Keskar, and Kishor M. Bhurchandi. Ball tracking in sports: A survey. *Artificial Intelligence Review*, 52(3):1655–1705, 2019. 10.1007/s10462-017-9582-2.

Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. 10.1002/nav.3800020109.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 10.1109/5.726791.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 10.1109/CVPR.2017.106.

Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021. 10.1007/s11263-020-01369-0.

Floriane Magera, Thomas Hoyoux, Olivier Barnich, and Marc Van Droogenbroeck. A universal protocol to benchmark camera calibration for sports. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3335–3346, 2024. 10.1109/CVPRW63382.2024.00338.

Wes McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010. 10.25080/Majora-92bf1922-00a.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 10.48550/arXiv.1912.01703.

Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 10.48550/arXiv.1804.02767.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 10.1109/CVPR.2016.91 .

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 91–99, 2015. 10.1109/TPAMI.2016.2577031 .

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016. 10.1109/CVPR.2016.89 .

Miguel Silva, Rui Marcelino, Daniel Lacerda, and Paulo Vicente Jo ao. Match analysis in volleyball: A systematic review. *Montenegrin Journal of Sports Science and Medicine*, 5(1):35–46, 2016.

Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020. 10.1016/j.imavis.2020.103910 .

Zoran Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 28–31, 2004. 10.1109/ICPR.2004.1333992 .