Player Tracking and Position Reconstruction in Football Game

Zekai Wang zw4603@nyu.edu

Abstract

The growing competitiveness of professional football has increased the demand for advanced sports analytics. Player tracking data, such as player movement and position, is a critical input for analysis pipelines. Traditional tracking systems rely on expensive hardware, such as wearable devices or high-end multi-camera setups, which provide high accuracy and real-time tracking, but impose significant financial and operational burdens on non-elite football clubs. Recent advances in computer vision offer a cost-effective alternative by enabling player tracking using standard video streams. This paper presents an integrated Game State Reconstruction (GSR) system that combines game reconstruction and jersey number detection into a unified pipeline. The system processes match videos to generate a 2D mini-map representing player movements and jersey numbers. This 2D view simplifies tactical analysis, enabling deeper insights into spatial dynamics while providing an intuitive visual summary of the game. My approach leverages publicly available pipelines and enhances them with modifications for improved accuracy and efficiency. By bridging the gap between high-cost tracking systems and affordable computer vision-based solutions, this project aims to empower football clubs, sports broadcasters, and analysts with accessible and cost-effective player tracking technology, opening new possibilities for tactical analysis and game review.

© 2025 Zekai Wang. All rights reserved.

1 Introduction

1.1 Business Potential and Competitiveness Analysis

The increasing commercialization of football and the intensification of competition among teams have elevated the demand for cost-effective player tracking solutions. Elite clubs traditionally rely on high-end systems such as Hawk-Eye and Catapult to monitor player movement, strategy, and performance. However, these systems impose significant financial barriers for smaller clubs and lower-tier leagues. Recent industry trends indicate a growing need for accessible alternatives, as evidenced by the rising popularity of platforms like Wyscout and Hudl[24]. To address this market gap, the Game State Reconstruction (GSR) system[18] offers an innovative software-driven approach that processes standard video streams, eliminating the need for costly, hardware-intensive setups. The GSR system produces 2D minimaps that visualize player positions, movements, and jersey numbers, facilitating intuitive tactical analysis. By leveraging computer vision and off-the-shelf cameras, the GSR system significantly lowers operational costs, enabling access to tracking technology for under-resourced football clubs and smaller leagues.

In contrast to traditional tracking systems that depend on proprietary hardware, the GSR system integrates player re-identification, jersey number recognition, and pitch calibration into a unified software pipeline. Competitors such as Catapult[1], Second Spectrum[19], and ChyronHego[2] rely on specialized equipment, while emerging computer vision-based (CV) solutions often focus on isolated aspects such as player or ball detection. The GSR system distinguishes itself by providing an end-to-end solution capable of processing standard video streams, creating a unique value proposition

Preprint. Under review.

for semi-professional teams and grassroots leagues. Furthermore, partnerships with leagues and teams offer opportunities for bespoke analytics dashboards tailored to specific client needs. By addressing challenges such as player occlusion and privacy regulations, the system's modular design ensures adaptability and scalability, supporting continuous feature updates and long-term market relevance. By prioritizing affordability, accessibility, and a comprehensive suite of tracking functionalities, the GSR system is positioned to become a pivotal force in the democratization of sports analytics.

1.2 Technical Overview

The Game State Reconstruction (GSR) relies on a modular pipeline architecture that integrates computer vision techniques[10] to ensure accurate player detection, role classification, team affiliation, and jersey number recognition.

Input and Preprocessing: The pipeline begins with the input of raw broadcast video footage. Frames are extracted from the video stream at a fixed rate and preprocessed to ensure compatibility with the downstream modules. This preprocessing includes resizing, format conversion, and basic frame normalization to improve the performance of the subsequent detection and tracking steps.

Athlete Detection and Tracking: Once the frames are prepared, the YOLOv8[9] model detects players within each frame. Only bounding boxes classified as "person" are retained. To maintain continuity across frames, the StrongSORT[4] multi-object tracking algorithm is applied, associating player detections across frames to form continuous player tracklets. The system tracks the position of each player in image coordinates and establishes player trajectories.

Pitch Localization and Camera Calibration: For pitch localization and camera calibration, the GSR system follows a geometry-based approach inspired by the work of Gutieerrez-Perez and Agudo[6]. The calibration process begins with the detection of keypoints and pitch lines using a convolutional neural network[13] (CNN) to generate heatmaps for key positions on the pitch. The camera parameters, including intrinsic and extrinsic properties, are estimated using a Direct Linear Transformation (DLT) algorithm, which maps 2D image coordinates to 3D pitch coordinates. A homography matrix is calculated to enable the projection of player positions from image space to pitch space. This calibration process ensures accurate localization of players on a 2D minimap while accounting for pitch geometry and camera orientation.

Player Identification and Role Classification: Once players have been localized on the pitch, player identification and role classification are performed. The PRTReID[15] model generates embeddings for each detected player, encoding features that help differentiate players based on role, team affiliation, and visual appearance. These embeddings are used to classify the role of each player (e.g., player, goalkeeper, referee) and assign team affiliation (e.g., left team, right team). The embeddings are also used to ensure that player identities remain consistent throughout the video, even if players disappear from view momentarily.

Jersey Number Recognition: The recognition pipeline design is inspired by an ingenious work[12]. Following athlete detection and tracking, each player's bounding box is used to crop player images from all frames. These cropped images are grouped into a tracklet corresponding to a single player (or object) with a unique track ID. To recognize jersey numbers from tracklet-level data, the system employs a refined pipeline. Each tracklet is processed through the following stages:

- Legibility Classification: A ResNet34-based[7] classifier identifies frames within the tracklet where the jersey number is clearly visible.
- Pose Detection: Keypoints corresponding to the player's torso are identified using a pose estimation model, and the torso region is cropped for subsequent processing.
- Jersey Number Recognition: The cropped torso images from each tracklet are processed using a set of five fine-tuned optical character recognition (OCR) models: SVTR-tiny (ResNet18), SVTR-small (ResNet34), SATRN (ResNet50), NRTR (ResNet101), and ASTER (ResNet152). Each model generates independent jersey number predictions for each frame in the tracklet.

© 2025 Zekai Wang. All rights reserved.

- Prediction Consolidation: A majority-vote mechanism is applied across all frame-level predictions for each tracklet, producing a robust, consolidated jersey number prediction at the tracklet level. This approach enhances robustness against occlusions, motion blur, and variability in frame quality. The final tracklet-level jersey number prediction is then reintegrated into the GSR system, replacing the previously used MMOCR-based method[20].

Tracklet Consistency and Cross-Level Reassignment: Since player role, team affiliation, and jersey numbers can vary across frames, a tracklet consistency check is performed. A cross-level reassignment algorithm is employed to handle two specific types of object reassignments. Image-level reassignment occurs when two bounding boxes exhibit a high intersection-over-union (IoU) and YOLO mistakenly identifies them as a single object. In such cases, the algorithm tracks objects that have disappeared and re-emerges with high IoU later, ensuring that the newly detected object is assigned its original track ID rather than being labeled as a new object. Tracklet-level reassignment is used to merge two tracklets if the system detects that they belong to the same player. By comparing the cosine similarity of embeddings and IoU of bounding boxes across tracklets, the algorithm merges tracklets associated with the same player but treated as separate due to tracking discontinuities. This process ensures consistency in player tracking and prevents unnecessary fragmentation of tracklets.

Output Generation and Visualization: Once the player's tracklets, roles, teams, and jersey numbers have been finalized, the system generates the final 2D minimap. The player positions, jersey numbers, and team affiliations are overlaid onto a scaled-down version of the football pitch. This minimap provides a clear, top-down view of the game, making it suitable for tactical analysis, performance reviews, and media broadcasts. The output can be visualized in real time or saved as a post-game analysis file, allowing users to replay key moments or analyze the movements of specific players over time.

© 2025 Zekai Wang. All rights reserved.

2 Related Work

2.1 Player Detection and Tracking

Player detection and tracking are fundamental to Game State Reconstruction. YOLOv8, known for its high-speed and accurate object detection, serves as the backbone for player detection. For tracking, StrongSORT extends the capabilities of DeepSORT[23] by incorporating appearance embeddings, enabling robust player tracking across frames. This combination ensures precise player localization and continuous tracking, even during occlusions or rapid movement.

2.2 Player Re-Identification and Role Classification

Maintaining player identity and role classification throughout a match is essential for tracking player activities. I employed PRTReID, a player re-identification framework that generates appearance-based embeddings for each player. By utilizing both appearance and temporal features, PRTReID enables consistent role classification (e.g., player, goalkeeper, or referee) and continuity of player identities, even when players are momentarily occluded. This approach ensures accurate player differentiation, which is critical for downstream analysis and track consistency.

2.3 Pitch Calibration and Homography Estimation

Accurate player localization on a 2D pitch requires precise pitch calibration. To achieve this, I leverage No Bells, Just Whistles Calibration, which detects pitch lines and key points using convolutional neural networks (CNNs) to generate heatmaps for critical pitch features. These features enable the estimation of camera parameters, including both intrinsic and extrinsic components, using a Direct Linear Transformation (DLT) algorithm. The homography matrix, computed through RANSAC and optimized via Levenberg-Marquardt, projects player positions from image coordinates to pitch coordinates, enabling precise player localization on a 2D minimap.

2.4 Jersey Number Recognition

Jersey number recognition is a key task in player identification, and I address it using a multi-model OCR approach. Five fine-tuned models form the backbone of this process:

- SVTR-tiny[3] (ResNet18): A lightweight OCR model optimized for efficiency, offering rapid processing of player images.
- SVTR-small[3] (ResNet34): A more robust version that improves accuracy, particularly for jerseys with partial occlusions.
- SATRN[14] (ResNet50): Known for its robust sequential text recognition, SATRN enhances recognition of challenging cases, such as skewed or partially visible jersey numbers.
- NRTR[16] (ResNet101): With a deeper backbone, NRTR is well-suited for complex jersey number configurations or difficult-to-read text.
- ASTER[17] (ResNet152): The most powerful of the five models, ASTER uses rectification techniques to normalize distorted text, significantly improving OCR performance in dynamic sports settings.

After frame-level predictions are generated by these models, a majority-vote mechanism consolidates frame-level predictions into a single tracklet-level result. This multi-model approach enhances robustness against occlusions, partial visibility, and motion blur, resulting in accurate jersey number recognition.

2.5 Pose Estimation for Jersey Number Detection

Pose estimation aids in jersey number recognition by focusing on the torso region, which typically contains the jersey number. I incorporate ViTPose[25], a Vision Transformer-based pose estimation model, to identify player keypoints. By cropping the torso area of players based on detected keypoints, I ensure that only the relevant area is passed to the OCR models. This step improves the accuracy and efficiency of jersey number recognition by reducing noise from irrelevant areas of the player's body.

2.6 Tracklet Consistency and Reassignment

Player tracking is susceptible to errors caused by tracking discontinuity and identity switches. To address this, I implement a cross-level reassignment algorithm. At the image level, track IDs are reassigned when a previously tracked player disappears and later reappears with high intersection-over-union (IoU) relative to an existing bounding box. At the tracklet level, player identities are merged if their embeddings have high cosine similarity and if their bounding boxes show consistent overlap across frames. This two-level reassignment process significantly improves the stability of tracking and maintains player identity, even in challenging scenarios.

© 2025 Zekai Wang. All rights reserved.

3 Implementation Details

In this section, I will mainly focus on the implementation details of modification I made towards the whole pipeline.

3.1 Environment Configuration For Inference And Finetune

Running the whole pipeline and finetuning models involves using High Performance Cluster (HPC) provided by the university. The standard running configuration is:

- GPU type: rtx8000
- Number of GPU: 2
- Number of CPU cores per task: 4
- Memory: 32G
- Maximum Running Time: 10Hours

3.2 Dataset Preparation For Model Finetuning

I used SoccerNet released the Jersey Number Recognition dataset, it is a large public jersey number recognition dataset. This dataset consists of a collection of player tracklets and contains tracklet-level annotation. The dataset is partitioned into training, test and challenge, with a total of 4,064 tracklets,

Table 1: Data set information

	Train	Test	Total
Tracklets	1,427	1,211	4,064
Images	733,000	564,500	204,600

but I will just need training and test set. The average length of tracklets is 482 frames. Table1 contains dataset statistics and figure 1 shows sample images.

Figure 1: Two sample images in the dataset

© 2025 Zekai Wang. All rights reserved.

3.3 Fine-tuning Models

Fine-tuning is a crucial step in optimizing pre-trained models for a specific task. For this project, I fine-tune five OCR models—SVTR-Tiny, SVTR-Small, SATRN, NRTR, and ASTER—to enhance their performance on jersey number recognition. Fine-tuning allows the models to adapt to the domain-specific data while leveraging the pre-trained knowledge from large-scale datasets.

3.3.1 Pre-Trained Models

These models were pre-trained on datasets such as MJSynth[8], SynthText[5], and other large-scale text recognition datasets. Their pre-trained weights provide a strong baseline to avoid training from scratch, reduce training time, and improve model generalization.

3.3.2 Data Preparation

The dataset consists of images of player jerseys, each labeled with the player's jersey number. The dataset includes a combination of manually labeled images and pseudo-labeled images filtered using a legibility classification model.

Valid Images: A JSON file contains paths of valid images to be used for fine-tuning.
Ground Truth Labels: Labels for each image are stored in a separate JSON file, where each image path is mapped to the corresponding jersey number.

Data Preprocessing Pipeline

1. Image Preprocessing: Images are resized to (32, 100) and normalized using mean and standard deviation values specific to the ImageNet dataset.
2. Augmentations: Data augmentations are applied to increase model robustness, including random rotation, random cropping, and color jittering.
3. Dataset Handling: Images and labels are combined using a PyTorch Custom Dataset class, which dynamically loads and processes images during training.

3.3.3 Model Architecture

SVTR-Tiny, SVTR-Small
Backbone: Convolutional layers followed by Local-Global Mixer blocks, see figure2.
Encoder: Uses attention-based transformer-like modules for robust feature extraction.

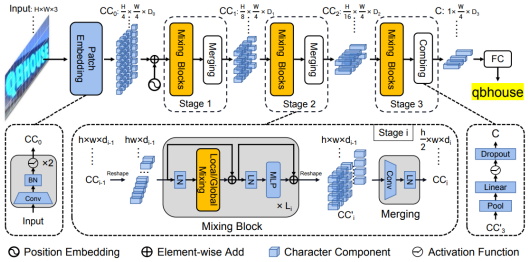

Figure 2: Architecture of SVTR

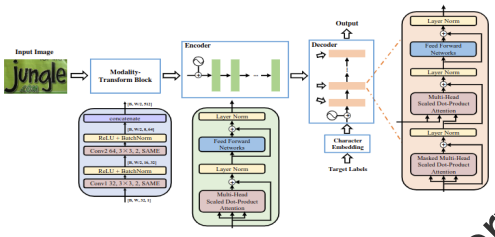

Figure 3: Architecture of NRTR

SATRN, NRTR, ASTER
Backbone: Transformer-based[21] encoder with positional embeddings, example of NRTR is shown at figure3.
Encoder: Multi-layer transformer blocks with self-attention mechanisms.

3.3.4 Fine-Tuning Strategy

Loading Pre-Trained Weights:
Pre-trained weights for the models were loaded using PyTorch's torch.load() function. Since pre-trained models often have key prefixes (e.g., 'module.' or 'state_dict.'), a utility function was developed to clean and align the keys in the state dictionary to match the model's parameter names.

Freezing Layers:
For some experiments, the early layers of the encoder were frozen to retain general visual features while only training the classification layer. This strategy reduces training time and avoids overfitting.

3.3.5 Training Configuration

See Table 2.

3.3.6 Training Process

- **Epochs:** All models were fine-tuned for 20 epochs.
- **Loss Function:** For classification models like SVTR-Small, a Cross-Entropy Loss was used to calculate the error between predicted and actual jersey numbers.
- **Training Loop:** Each epoch consists of a forward pass, backward pass, and gradient update.

Table 2: Training Configuration

Batch size	32
No. of workers	4
Optimizer	Adam optimizer
Learning rate	1e-4
Weight decay	0.05
Scheduler	Cosine Annealing Scheduler

© 2025 Zeka Wang. All rights reserved.

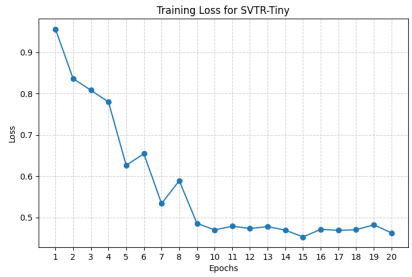

Figure 4

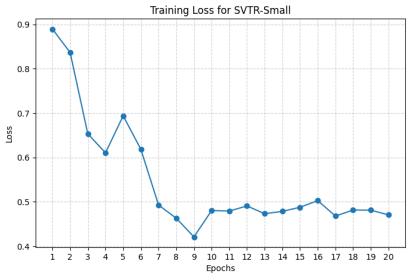

Figure 5

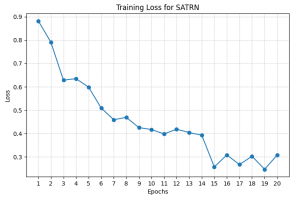

Figure 6

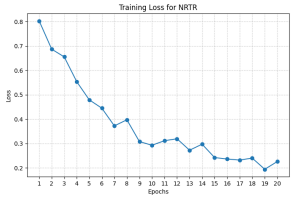

Figure 7

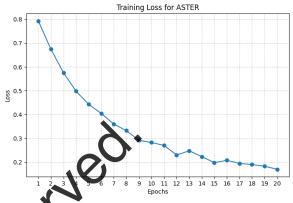

Figure 8

Figure 9: Training loss of 5 models

© 2025 Zekai Wang. All rights reserved.

3.3.7 Validation and Testing

During training, model performance was validated on a subset of data to monitor overfitting. The evaluation metrics included: - Accuracy: The percentage of correctly predicted jersey numbers.

The final evaluation was conducted on the test set, which was separate from the training and validation data.

3.3.8 Challenges and Solutions

Challenges:

1. Imbalanced Data: Some jersey numbers appeared more frequently than others, causing the model to favor these classes.
2. Noise in Pseudo-Labels: Pseudo-labeled images introduced incorrect labels due to OCR errors.

Solutions:

1. Data Augmentation: Introduced random rotations, color jittering, and Gaussian noise to make the model more robust.
2. Dynamic Batch Size: Reduced batch size for larger models and used gradient accumulation to simulate larger batch sizes.

3.3.9 Results

See Table 3.

3.3.10 Integration Pipeline

The fully pipeline with specified modules:
YOLOv8 -> PRTReId -> StrongSORT -> NBJW_Calib_Keypoints -> NBJW_Calib -> Jersey Number Recognition (originally MMOCR) -> MajorityVoteTracklet -> Reassignment Algorithm -> Tracklet-TeamClustering -> TrackletTeamSideLabeling

Model	Validation Accuracy	Test Accuracy	Training Time
SVTR-Tiny	65.3%	64.9%	2 hours
SVTR-Small	68.7%	68.1%	2.5 hours
SATRN	70.2%	69.7%	3 hours
NRTR	69.8%	69.2%	3 hours
ASTER	71.1%	70.8%	4 hours

Table 3: 5 models training results

The output of the "jersey number recognition" is in .json format like this: "1": "19", "26": "10", "29": "-1".... For example, player with track_id == 1 is predicted to have jersey number of "19", player with track_id == 29 is predicted that no jersey number is visible.

Then, the function add_jersey_numbers_to_csv() will append the jersey number prediction result to the output of module "NBJW_Calib" and skip the original MMOCR module, moving forward to the next module.

3.4 Cross Level Reassignment

Maintaining the overall tracking consistency is also crucial, since inconsistency in object tracking will affect the jersey number recognition too. Therefore, to explicitly help with this issue, I developed this tiny algorithm to "recover" lost track objects.

3.4.1 Frame Level Reassignment

The frame-level object reassignment algorithm addresses bounding box misidentification caused by overlapping or closely positioned detections within a single frame. By leveraging the Intersection over Union (IoU) metric, the algorithm dynamically evaluates pairwise IoU values and resolves conflicts to maintain consistent object identities. See figure 10.

IoU (Intersection over Union): Measures spatial overlap between bounding boxes of two tracklets.
ID_Dict: Maintains mappings between bounding boxes (bbox) and their associated bounding boxes.
IOU_Dict: Stores IoU values for bounding box pairs.
IOU_THRESHOLD: Determines sufficient spatial overlap for reassignment.

3.4.2 Tracklet Level Reassignment

The tracklet-level reassignment algorithm resolves discontinuities in player tracking by merging tracklets that correspond to the same object. It evaluates spatial and appearance similarity between pairs of tracklets to ensure robust identity maintenance. See figure 11.

Cosine Similarity: Measures the feature similarity between tracklets using embedding vectors.
IOU_THRESHOLD: Minimum IoU required to consider two tracklets spatially consistent.
SIM_THRESHOLD: Minimum cosine similarity for two tracklets to be considered visually similar.

© 2025 Zekai Wang. All rights reserved.

```
// Frame level reassignment
ID_Dict = []
IOU_Dict = []
FOR each bbox1 IN detections:
   FOR each bbox2 IN detections:
       iou = compute_iou(bbox1, bbox2)
       IF iou > IOU_THRESHOLD
              IF bbox1 exists in ID_Dict.keys:
                     old_pair = ID_Dict("bbox1")
                     IF IOU_Dict(bbox1, old_pair) < iou
                            // bbox2 is likely used to paired
                     with bbox1
              Elif bbox2 exists in Dict.keys:
                     old_pair = ID_Dict("bbox2")
                     IF IOU_Dict(old_pair, bbox2) < iou
                            // bbox1 is likely used to paired
                     with bbox2
              Elif bbox1, bbox2 exist in ID_Dict.keys
                     // Do nothing
              ELSE:
                     ID_Dict("bbox1": bbox2, "bbox2": bbox1)
                     IOU_Dict((bbox1, bbox2), iou)
```

Figure 10: Pseudo code of frame level reassignment

```
// Tracklet level reassignment
FOR each tracklet1 IN tracklets:
   FOR each tracklet2 IN tracklets:
       iou = compute_iou(tracklet1, tracklet2)
       sim = cosine_sim(tracklet1, tracklet2)
       IF iou > IOU_THRESHOLD AND sim > SIM_THRESHOLD
        MERGE tracklet1 and tracklet2
```

Figure 11: Pseudo code of tracklet level reassignment

© 2025 Zekai Wang. All rights reserved.

4 Results and Analysis

4.1 Introduction to Evaluation and Metrics

The evaluation of the proposed GSR pipeline is conducted to measure its overall performance and analyze the contributions of individual components. The primary metric used is Global Sports Higher Order Tracking Accuracy (GS-HOTA), which provides a comprehensive assessment by combining detection accuracy, track association, and temporal re-identification consistency. Sub-metrics such as detection accuracy, track association accuracy, and localization accuracy are also analyzed to identify performance bottlenecks. This section presents the evaluation results and highlights areas of strength and potential improvement.

4.2 Pipeline Performance Using GS-HOTA

The performance of the entire pipeline is evaluated across 15 different game video clips using the GS-HOTA metric and other sub-metrics, which captures the system's ability to accurately detect,

$$Sim_{\text{GS-HOTA}}(P, G) = \text{LocSim}(P, G) \times \text{IdSim}(P, G),$$

$$\text{IdSim}(P, G) = \begin{cases} 1 & \text{if all attributes match,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{with LocSim}(P, G) = e^{\ln(0.05)\frac{\|P-G\|_2^2}{\tau^2}},$$

Figure 12: GS-HOTA

Metric	Score (On Average)	Sub-metric	Score (On Average)
GS-HOTA	0.4	DetRe	0.5
DetA	0.3	DetPre	0.5
AssA	0.5	AssRe	0.6
LocA	0.8	AssPre	0.8

Table 4: Evaluation result, the score is round-off to 1 sig.fig

© 2025 Zekai Wang. All rights reserved.

track, and maintain consistent player identities across video frames. The evaluation is summarized in Table4.

The overall performance score of 0.4 shows that it is a good starting point, but there are many rooms for improvement. The results demonstrate high localization accuracy (LocA), indicating that the YOLOv8-based detection module effectively identifies player bounding boxes with minimal false positives. However, the relatively lower scores in detection accuracy (DetA) and association accuracy (AssA) is observed.

4.3 Result Analysis and Bottleneck Breakdown

To identify bottlenecks, I analyze the result on sub-metrics and the contributions of individual components:

- Jersey Number Detection Precision (DetPre) and Recall(DetRe): Many misclassifications occur in dense regions where multiple players overlap and affect the detection precision. Also, false positives occurs quite often since the detection pipeline is still under-performing in real games, which lowers the recall score. One potential reason might be the trained models are still not generalized enough to identify non-number pictures. Also, one very important reason is that, the evaluation method does not count for correct even the detected jersey number is if the player has been tracked wrongly (identified as a new object)

- Player Tracking Association Precision (AssPre) and Recall(AssRe): Track association recall (0.6) reveals instances of identity switches caused by occlusions or overlapping trajectories. These errors occur primarily in crowded areas where players temporarily disappear or reappear. The frame-level reassignment algorithm mitigates this issue by re-evaluating bounding boxes based on IoU, since the precision is relatively satisfactory; however, some errors persist due to rapid player movement.

The analysis highlights that the primary bottlenecks arise during tracking and re-identification(re-tracking players to maintain association) and the generalization power of the jersey number detection pipeline, particularly in crowded or ambiguous player interactions. Future work can address these issues by enhancing the reassignment algorithms or refining the detection pipeline. Also, poor calibration might also affect the score, but it requires further investigation.

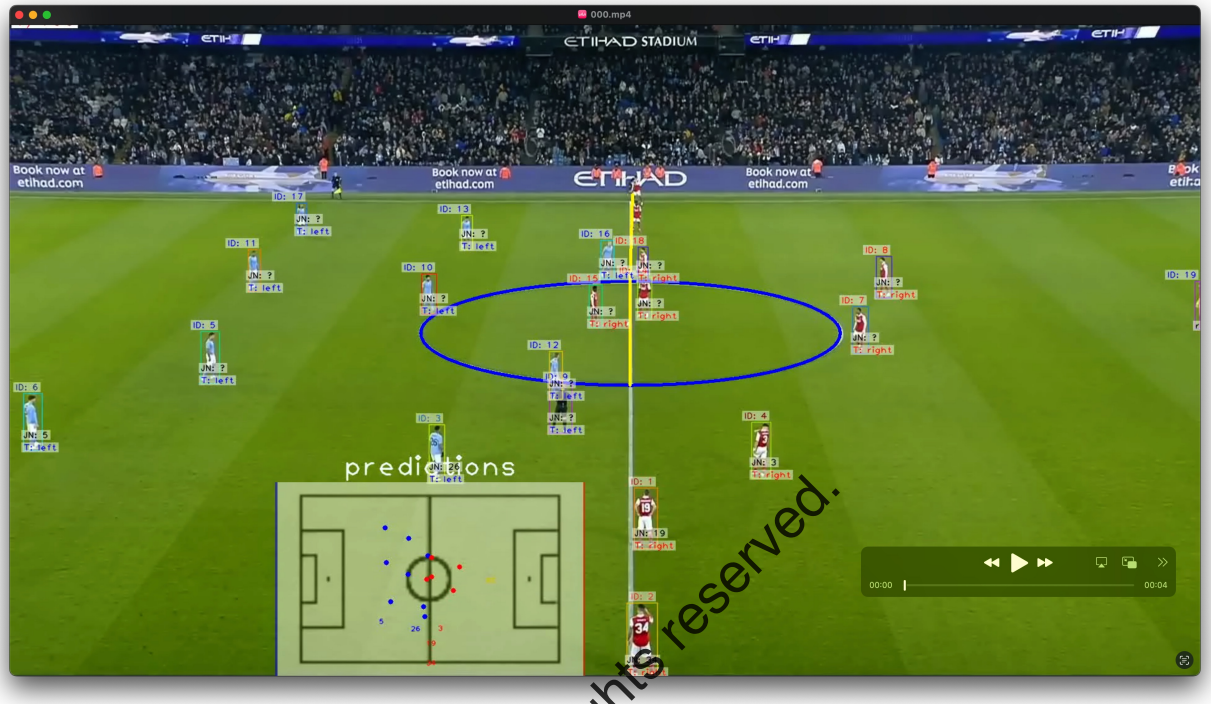

Figure 13: Good Performance Example

© 2025 Zekai Wang. All rights reserved.

4.4 Qualitative Analysis

To complement the quantitative results, I provide visual examples that illustrate the pipeline's performance:

Successful Cases
In figure 13, it has relatively clear visibility and minimal occlusions, the pipeline achieves accurate player detection, jersey number recognition, and consistent tracking. The generated 2D minimaps effectively represent player positions, movements, and roles.

Failure Cases: Errors are observed in challenging scenarios: Identity Switches: In the consecutive frames in figure 14 and figure 15, when two players overlap or occlude each other, the tracking module occasionally assigns incorrect identities, leading to fragmented tracklets.
Jersey Number Misrecognition: Partial visibility, motion blur, or torso misalignment during pose estimation can result in incorrect jersey number predictions.

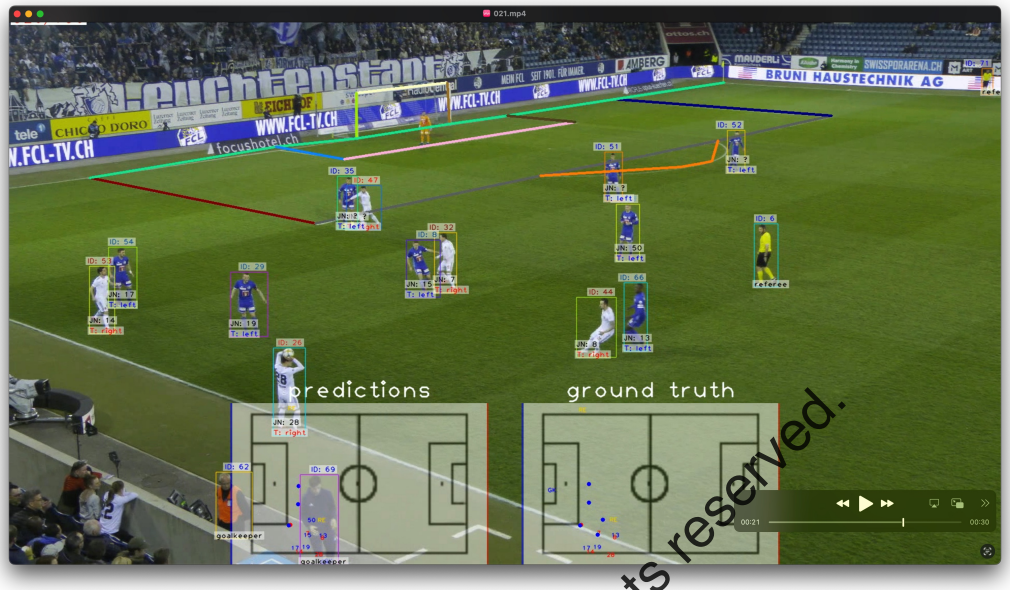

Figure 14: Bad Performance Example 1

© 2025 Zekai Wang. All rights reserved.

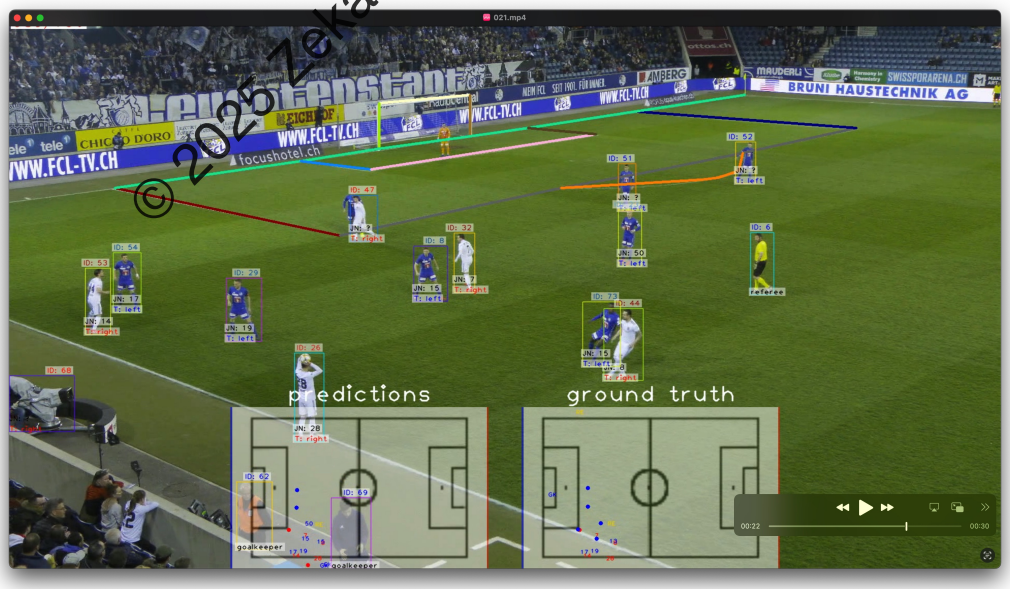

Figure 15: Bad Performance Example 2

5 Conclusions

This project successfully develops a beginner-level solution for Game State Reconstruction (GSR), enabling the generation of 2D minimaps of football matches from broadcast video streams. The pipeline integrates key modules for player detection, tracking, re-identification, and jersey number recognition, providing a cost-effective, software-driven alternative to hardware-dependent tracking solutions.

A major achievement of this project is the integration of an advanced jersey number detection module, incorporating self-modifications into the original GSR pipeline. Additionally, a cross-level reassignment mechanism is introduced to maintain object tracking consistency and accuracy at both the frame and tracklet levels.

The implementation of the jersey number recognition module combines modern OCR techniques, employing five fine-tuned models—SVTR-tiny, SVTR-small, SATRN, NRTR, and ASTER—with a majority-vote mechanism to enhance accuracy. ViTPose further refines the player torso region, improving OCR input for precise jersey number recognition, especially under conditions of occlusion or motion blur.

Evaluation of the pipeline using the GS-HOTA metric demonstrates its potential, with an overall average GS-HOTA score of 0.4, detection accuracy of 0.3, association accuracy of 0.5 and localization accuracy of 0.8. So there are potential improvement in dense player interactions. Enhancements in detection models and reassignment logic are suggested to further reduce identity switches.

In conclusion, this project introduces a some enhancement to the original Game State Reconstruction framework. Ongoing work will focus on enhancing tracking robustness, optimizing processing efficiency, and ensuring system readiness for real-time applications.

© 2025 Zekai Wang. All rights reserved.

6 Future Work

The development of the Game State Reconstruction (GSR) system has established a solid foundation for advanced football match analysis using broadcast video streams. To further enhance its performance, future work will focus on Structural Improvements and Modular Improvements.

6.1 Structural Improvements

6.1.1 Latency Reduction via Key Frame Extraction

The current GSR pipeline processes all frames, resulting in high computational costs. Key frame extraction aims to reduce system latency by processing only frames with significant changes in player movements or jersey visibility. Techniques such as frame-difference analysis and pose change detection can be employed to identify key frames. This approach will enable real-time processing while maintaining essential game information.

6.1.2 Video-Level Consistency

Maintaining consistent player identities throughout a video is a challenge, especially when players leave and re-enter the frame or when occlusions occur. Current frame-level and tracklet-level consistency methods will be expanded to video-level consistency using long-term memory and video-spanning embeddings. These techniques will ensure persistent player identities across the entire video, reducing identity switches and enhancing tracking stability.

6.2 Modular Improvements

6.2.1 Enhanced Pitch Calibration Using a CNN-Transformer Hybrid

Current pitch calibration relies on feature detection to compute homography transformations. Future work will introduce a CNN-Transformer hybrid model[22] to improve feature extraction and geometric transformation. CNNs will handle local feature extraction, while Transformers will capture global pitch context, resulting in more accurate player localization, even in challenging broadcast conditions.

6.2.2 Time-Aware Training for Sequential Information

Current player tracking and jersey number recognition models operate on frame-level data without explicit use of time-sequence information. Incorporating temporal models such as Recurrent Neural Networks[11] (RNNs) or Transformer-based temporal models will enable the system to leverage temporal dependencies between frames. This improvement will enhance re-identification and jersey number recognition by utilizing player motion patterns, thereby reducing errors from occlusions or sudden reappearances.

The proposed enhancements aim to improve efficiency, accuracy, and robustness, enabling real-time performance, better player tracking, and seamless identity management across entire football matches.

7 References

References

[1] Catapult. Catapult official website. `https://www.catapult.com/`. Accessed: December 2024.

[2] ChyronHego. Chyronhego official website. `https://chyronhego.com/`. Accessed: December 2024.

[3] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model, 2022.

[4] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

[5] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] Marc Gutiérrez-Pérez and Antonio Agudo. No bells just whistles: Sports field registration by leveraging geometric properties. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3325–3334, June 2024.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[8] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European Conference on Computer Vision*, 2014.

[9] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[10] Victor Joos, Vladimir Somers, and Baptiste Standaert. TrackLab. `https://github.com/TrackingLaboratory/tracklab`, 2024.

[11] M I Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. 5 1986.

[12] Maria Koshkina and James H. Elder. A general framework for jersey number recognition in sports video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3235–3244, June 2024.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[14] Junyeop Lee, Sungrae Park, Jeonghun Baek, Seong Joon Oh, Seonghyeon Kim, and Hwalsuk Lee. On recognizing texts of arbitrary shapes with 2d self-attention, 2019.

[15] Amir M. Mansourian, Vladimir Somers, Christophe De Vleeschouwer, and Shohreh Kasaei. Multi-task learning for joint re-identification, team affiliation, and role classification for sports visual tracking. page 103–112. ACM, October 2023.

© 2025 Zekai Wang. All rights reserved.

[16] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition, 2019.

[17] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2035–2048, 2019.

[18] Vladimir Somers, Victor Joos, Silvio Giancola, Anthony Cioppa, Seyed Abolfazl Ghasemzadeh, Floriane Magera, Baptiste Standaert, Amir Mohammad Mansourian, Xin Zhou, Shohreh Kasaei, Bernard Ghanem, Alexandre Alahi, Marc Van Droogenbroeck, and Christophe De Vleeschouwer. SoccerNet game state reconstruction: End-to-end athlete tracking and identification on a minimap. June 2024.

[19] Second Spectrum. Second spectrum official website. `https://www.secondspectrum.com/index.html`. Accessed: December 2024.

[20] MMOCR Developer Team. Mmocr: A comprehensive toolbox for text detection, recognition and understanding. 2022.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[22] Lingfei Wang, Chenghao Zhang, and Jin Li. A hybrid cnn-transformer model for predicting n staging and survival in non-small cell lung cancer patients based on ct-scan. *Tomography*, 10(10):1676–1693, 2024.

[23] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017.

[24] Wyscout. Wyscout login page. `https://identity.hudl.com/u/login/identifier?state=hKFo2SBIVEhmMGFvTUNpQUZtUVdsTnN2cm0zb2FzOHVGelZUdKFur3VuaXZlcnNhbC1sb2dpbqN0aWTZIDI5b1V` Accessed: December 2024.

[25] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022.

© 2025 Zekai Wang. All rights reserved.