Computer Vision and Image Understanding 263 (2026) 104602

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

An end-to-end pipeline for team-aware, pose-aligned augmented reality in cycling broadcasts☆

Winter Clinckemaillie *, Jelle Vanhaeverbeke, Maarten Slembrouck, Steven Verstockt

IDLab, Ghent University-imec, Ghent, 9052, Belgium

ARTICLE INFO

Keywords:
Computer vision
Object detection
One-shot classification
3D pose estimation
Augmented reality
Sports broadcasting

ABSTRACT

Advanced computer vision and machine learning technologies transform how we experience sports events. This work enriches helicopter footage of cycling races with dynamic, in-scene, pose-aligned augmented reality (AR) overlays (e.g., rider name, speed, wind direction) that remain visually attached to each rider. To achieve this, we propose a multi-stage pipeline: cyclists are first detected and tracked, followed by team recognition using a one-shot learning approach based on Siamese neural networks, which achieves a classification accuracy of 85% on a test set composed of unseen teams during training. This design allows easy adaptation and reuse across different races and seasons, enabling frequent jersey and team changes with minimal effort. We introduce a pose-based AR overlay that anchors rider labels to moving cyclists without fixed field landmarks or homography, enabling dynamic overlays in unconstrained cycling broadcasts. Real-time feasibility is demonstrated through runtime profiling and TensorRT optimizations. Finally, a user study evaluates the readability, informativeness, visual stability, and engagement of our AR-enhanced broadcasts. The combination of advanced computer vision, AR, and user-centered evaluation showcases new possibilities for improving live sports broadcasts, particularly in challenging environments like road cycling.

1. Introduction

Cycling, in particular, has lagged behind other sports in adopting interactive and dynamic broadcast technologies and faces growing pressure to enhance its traditional broadcast model to meet modern viewer demands. Despite advancements in broadcasting technology, coverage of cycling events remains largely manual. Key information, such as rider names and team affiliations, is still inserted manually, a workflow that scales poorly and limits the depth of insights provided to viewers. In contrast, sports like football and basketball utilize augmented reality (AR) for dynamic overlays and real-time statistics (Goebert, 2020). For example, in football, virtual advertising dynamically adjusts digital banners based on the viewer's location or the target market of the broadcast. Technologies like these enable broadcasters to tailor advertising content during live broadcasts, providing localized and personalized experiences for viewers (Supponor, 2020). However, applying similar AR concepts to cycling is much more challenging due to the unique nature of its broadcasting environment. Unlike stadium sports with controlled conditions, cycling races take place outdoors over vast terrains and lack fixed field landmarks or per-venue camera calibration. These challenges complicate the implementation of automated, dynamic overlays that remain consistent throughout the race.

This paper aims to address these challenges by exploring how advanced computer vision and artificial intelligence (AI) techniques can enhance cycling broadcasts, making them more interactive and informative. The first part focuses on cyclist detection, tracking, and team recognition in aerial cycling footage, a task complicated by occlusions from varying camera angles, such as side or frontal views, and visual obstructions like trees. Additionally, changing lighting conditions and the visual similarity of team jerseys make this task even more complicated. This work presents team recognition as a one-shot learning task using a Siamese network, improving accuracy and efficiency while generalizing across different races and evolving jerseys without the need for extensive retraining.

The second part introduces a pose-based AR overlay method that anchors overlays to moving cyclists without relying on fixed landmarks or homography, enabling in-scene, rider-specific visualizations even in uncontrolled cycling broadcasts. This is achieved by estimating a local, rider-centric 3D anchor (an oriented 3D box derived from monocular 3D pose estimation), which allows labels and markers to remain visually attached to each cyclist as the camera moves or zooms. Fig. 1 illustrates examples of these AR elements, including dynamic overlays that follow riders and display relevant data during the race.

☆ This article is part of a Special issue entitled: 'CV for Sports' published in Computer Vision and Image Understanding.

* Corresponding author.
E-mail address: winter.clinckemaillie@ugent.be (W. Clinckemaillie).

https://doi.org/10.1016/j.cviu.2025.104602
Received 19 September 2025; Received in revised form 5 November 2025; Accepted 2 December 2025
Available online 4 December 2025

Fig. 1. Examples of augmented reality overlays applied to helicopter footage during cycling races. The static card overlay (left) displays rider data. The dynamic name card (middle) follows the rider's movement and orientation. The ground marker (right) projects a visual indicator on the ground beneath the cyclist.

By combining team recognition with AR, this work aims to enhance cycling broadcasts, reducing manual input and improving race coverage with contextual visualizations.

This work builds on initial results presented in Clinckemaillie et al. (2026). The main contributions of this work are:

- A one-shot team recognition approach that generalizes to unseen jerseys using a Siamese neural network.
- *CyclingTrack* benchmark: dataset for cyclist tracking in broadcast footage, with a comparative evaluation of recent multi-object tracking methods.
- A pose-based AR overlay method enabling dynamic, rider-specific visualizations in unconstrained environments, using a rider-centric 3D anchor. This method has been assessed against conventional overlays in a user study.
- An end-to-end pipeline for automated cyclist detection, tracking, team recognition, and AR overlay, with runtime optimizations for real-time feasibility.

In Section 2, we review related work on cyclist recognition and augmented reality applications in sports broadcasts. Section 3 outlines the datasets and data collection procedures used in this study. In Section 4, we describe our proposed pipeline, which includes detection, tracking, team recognition, 3D bounding box calculation, and AR visualizations. The main experiments and results are presented in Section 5. Section 6 reports the user experience study of our proposed AR overlays. Section 7 summarizes key limitations of our approach, and Section 8 concludes with directions for future research.

2. Related work

This section provides an overview of the scientific context for this research, focusing on two main areas: identifying cycling teams and utilizing dynamic AR elements in sports broadcasts. This sets the foundation for the research gaps addressed in this work, highlighting existing knowledge and ongoing challenges.

2.1. Cyclist detection and team recognition

Cyclist detection. Accurate detection of cyclists is crucial for analyzing race footage, as it forms the foundation for tracking and team recognition. This requires high accuracy and speed, especially for real-time applications. Challenges include distinguishing cyclists from motorcycles and spectators, managing occlusions within the peloton, and dealing with varying camera perspectives (Naik et al., 2022). Recent deep learning approaches, particularly convolutional neural networks (CNNs), have significantly improved detection accuracy. Among these, the YOLO (You Only Look Once) framework stands out for its ability to process entire images in a single forward pass, making it well-suited for real-time applications. YOLOv8 introduced an anchor-free architecture and improved feature extraction, enhancing small-object detection (Jocher et al., 2023). More recently, YOLOv11 further improved performance with attention and architectural refinements, offering a competitive accuracy–latency trade-off for real-time detection (Jocher and Qiu, 2024).

Transformer-based detectors have also advanced object detection. The DETR (Detection Transformer) family introduces set-based prediction and bipartite matching instead of anchors, but typically requires long training and can struggle on small objects such as cyclists (Carion et al., 2020). Deformable DETR improves both convergence and small-object performance using multi-scale deformable attention, achieving comparable accuracy in approximately ten times fewer epochs (Zhu et al., 2021). Recently, RF-DETR (Robinson et al., 2025) has emerged as a practical, real-time DETR variant designed for fine-tuning across diverse domains, and is increasingly used as a competitive baseline alongside YOLO in applied comparisons (Robinson et al., 2025).

In sports video analysis, methods such as time-sharing memory networks and skeleton-motion enhancement improve athlete detection by strengthening pose-aware temporal features (Ren, 2022). In cyclocross, where athletes alternate between cycling and running depending on terrain, a YOLOv5-based method successfully distinguishes these modes and produces valuable metadata for both live broadcast and post-race analysis (De Bock and Verstockt, 2021). While datasets on racing cyclists are limited, large benchmarks like EuroCity Persons (Braun et al., 2018) and the Tsinghua–Daimler Cyclist Benchmark (Li et al., 2016) provide annotations for cyclists in urban settings. However, these focus on ground-level views of larger, sparse groups, whereas our focus is on small, aerial views of dense pelotons. This underscores the need for broadcast-specific datasets featuring small, densely grouped cyclists in aerial views to support reliable detection.

Team recognition. After detecting cyclists, identifying them into their respective teams is essential for enriching live broadcasts and improving race analyses. Various methods have been explored in literature, each with specific applications and limitations in cycling. Liu and Bhanu introduced a pose-guided R-CNN for recognizing jersey numbers in soccer (Liu and Bhanu, 2019). While effective under controlled conditions, it is less suitable for aerial footage, where jersey numbers are rarely visible. Zhang et al. developed the DeepPlayer model, integrating Cascade Mask R-CNN for initial player detection along with a specialized component for accurate jersey number recognition (Zhang et al., 2020). The PoseID component of this model enhances identification by linking advanced representations of players to their specific body postures. However, variations in visibility during helicopter footage can limit its effectiveness. Verstockt et al. proposed a methodology using skeleton-based pose detection for identifying cyclists, combining pose orientation with jersey number recognition to improve identification accuracy (Verstockt et al., 2020). By leveraging both the structural data from cyclists' poses and visual cues from their jerseys, this dual focus

allows for more reliable team recognition, especially in dynamic scenes where traditional methods might struggle.

Siamese neural networks are well-suited to one-shot learning when per-class data are scarce (Duque Domingo et al., 2021; Koch, 2015). By learning to extract feature vectors from input images (such as team jerseys) and comparing them to a database of known examples, these networks minimize the distance between similar classes and maximize it between dissimilar ones using loss functions like contrastive loss or triplet loss (Schroff et al., 2015). This strategy allows for robust recognition of new cycling teams using just a single reference image per class, offering high adaptability and efficiency for team identification. Recent studies in sports, such as team recognition in soccer, further demonstrate the suitability of Siamese networks for visual jersey identification (Santos and Jerri, 2023).

Tracking. Accurate tracking of cyclists during broadcasts depends on the system's ability to remain robust under conditions involving moving cameras, dense pelotons, and frequent occlusions. Kalman filter-based online trackers, such as SORT (Simple Online and Realtime Tracking), propagate detections across frames by using the Hungarian assignment method, allowing them to bridge short gaps caused by obstacles like poles or trees (Bewley et al., 2016). However, identity switches increase as camera motion and crowding intensify. DeepSORT addresses this issue by incorporating a deep appearance descriptor into the association step (Wojke et al., 2017).

Recent online tracking-by-detection methods refine these ideas in complementary ways. ByteTrack improves recall by also associating low-score detections (Zhang et al., 2022). OC-SORT revisits SORT's update scheme with observation-centric updates to reduce error accumulation during occlusion and non-linear motion (Cao et al., 2023). StrongSort and BoT-SORT further combine motion with learned appearance embeddings (Aharon et al., 2022; Du et al., 2023). BoT-SORT also includes camera-motion compensation. Variants like BoT-SORT-ReID and Deep OC-SORT explicitly integrate re-identification to maintain consistent identities over longer gaps, at some extra computational cost (Maggiolino et al., 2023). BoostTrack reasons over tracklets and boosts detection confidence to stabilize assignments in crowded scenes, typically trading speed for robustness (Stanojevic and Todorovic, 2024).

However, existing benchmark datasets, such as MOTChallenge (MOT17, MOT20) and SportsMOT, mainly focus on controlled environments, including indoor or stadium-based sports captured by static cameras (Cui et al., 2023; Dendorfer et al., 2020). These settings are fundamentally different from the dynamic and outdoor nature of cycling broadcasts, which involve fast-moving cameras, large pelotons, frequent occlusions, and highly variable backgrounds. This distinction highlights a significant gap in datasets specifically designed to evaluate tracking performance under realistic cycling broadcast conditions.

Research gap. Although methods like YOLO, Siamese networks, and recent multi-object trackers show promising performance individually, their practical application in cycling broadcasts remains limited. Most were designed for ground-level or controlled settings and struggle with aerial views, small cyclists, and extensive occlusion. This paper aims to bridge the gap between existing methods and their practical usability in cycling broadcasts by adapting cyclist detection, one-shot team recognition, and tracking techniques specifically for these challenging conditions. In addition, we introduce a dedicated cyclist-tracking benchmark dataset tailored to aerial footage and dense pelotons, enabling reproducible evaluation and fair comparison under realistic broadcast constraints.

2.2. AR visualization

AR has transformed sports broadcasting by enhancing viewer experiences with dynamic overlays. For instance, in Formula (1) racing, AR displays real-time information such as drivers' names and positions directly above their cars (Noble, 2023). Similarly, AR overlays like the digital first-down line in American football have become standard, providing viewers with clear visual indicators of game progress and player positioning (Mashable, 2022). In tennis, systems like Hawk-Eye project ball trajectories onto the court, relying on precisely calibrated, high-speed cameras in controlled environments (Owens et al., 2003).

However, these systems are not directly applicable to cycling broadcasts. Unlike stadium sports or events in confined, well-defined areas, cycling races take place over long, open courses without fixed field landmarks or per-venue camera calibration. This makes traditional AR alignment methods, which often depend on global scene references (e.g., field lines or homography), less viable. In cycling, AR overlays must remain geometrically attached to moving athletes without relying on fixed scene references.

A calibration-free direction is to recover 3D human geometry from monocular broadcast footage, enabling body-aligned placement that follows each cyclist. 3D Human Pose Estimation (HPE) models predict 3D coordinates of key joints from 2D detections, while Human Mesh Recovery (HMR) estimates a full 3D mesh of the human body using a parametric model. Early single-image HMR and occlusion-aware variants (e.g., HMR (Kanazawa et al., 2018), PARE (Kocabas et al., 2021)) established this paradigm; more recently TokenHMR introduced a discrete token-based pose representation that improves in-the-wild accuracy from a single image (Dwivedi et al., 2024), which is attractive in broadcast contexts that typically offer only monocular helicopter views (see Fig. 2).

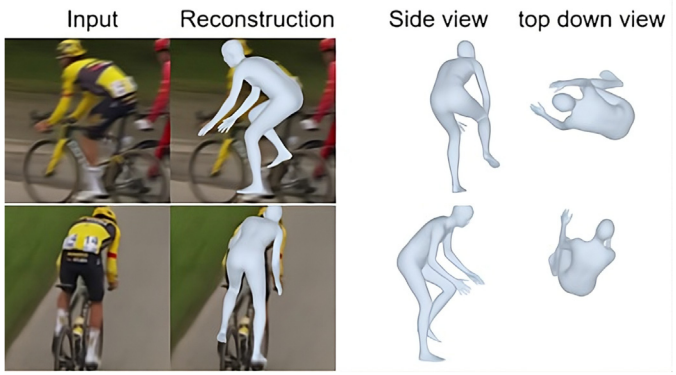


Fig. 2. Examples of reconstructed 3D meshes using TokenHMR from single RGB images.

Despite this progress, mesh models do not guarantee anatomically consistent shapes or the optimal joint localization in sports footage. Recent studies (Ludwig et al., 2025a,b) show that HMR models can drift in body shape across frames, and that anthropometric constraints or 2d-to-3d uplifting with joint rotations can further improve consistency and accuracy in dynamic, athletic scenes. Consequently, integrating HMR into AR for uncontrolled broadcasts requires additional considerations beyond raw mesh estimates, including robustness under occlusion, temporal stability, and computational efficiency for live use.

Research gap. While existing AR systems perform well in controlled environments, cycling broadcasts require calibration-free methods that remain reliable in the wild. We focus on how mesh-based 3D pose estimation can be operationalized for AR overlay placement in broadcast cycling, aiming for robust and anatomically coherent alignment that maintains accuracy and temporal stability under camera/rider motion and short occlusions. Addressing these challenges is crucial for deploying AR in cycling and offering viewers a clearer, more informative, and engaging broadcast experience.

3. Datasets

This section describes the datasets developed for cyclist detection, tracking and team recognition. Details of model training are given in Section 4, and quantitative results are reported in Section 5.

3.1. Cyclist detection

We created a custom dataset of annotated images for detecting cyclists and motorcycles from helicopter footage, available on Roboflow.[1] This dataset contains 660 images with 20,111 annotations, focusing on cyclists' upper bodies to enable team recognition by jerseys and logos. It also includes motorcycles to help differentiate cyclists from other objects commonly present in race footage. The dataset is split into training (70%), validation (15%), and test (15%) sets for thorough evaluation.

3.2. Team recognition (Jersey crops)

For team recognition, we constructed a custom dataset of cropped images of cyclist jerseys. Each crop measures at least 35 × 35 pixels and is resized to 105 × 105 pixels. Each image is labeled by team, and we applied data augmentation techniques, such as rotation and brightness adjustments, to enhance robustness, as illustrated in Fig. 3. The primary dataset includes an average of 80 images per team across 24 teams (with 70% for training and 30% for validation). To evaluate the one-shot capability, we established a separate test set comprising 15 new teams, each with an average of 10 images. This set assesses the recognition of unfamiliar jerseys, reflecting the frequent outfit changes in cycling.

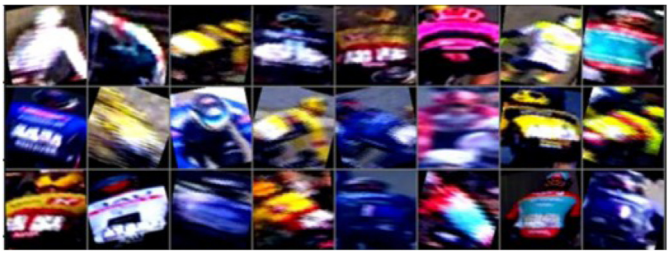


Fig. 3. Examples of cropped rider jersey images used for team recognition.

3.3. CyclingTrack

To benchmark and evaluate the performance of state-of-the-art tracking methods in cycling broadcasts, we introduce the *CyclingTrack* dataset.[2] It comprises 10 manually annotated video sequences of aerial footage from real-world races, recorded at 1920 × 1080 resolution and 25 frames per second. Sequences were selected to span diverse viewpoints and difficulty (front/side/top-down, sparse/dense peloton, varied occlusions), and the train/test split balances these factors.

Each sequence covers a distinct group or camera viewpoint for as long as that group is visible in the helicopter footage. This results in varying sequence lengths (from 7 to 85 s), reflecting how actual cycling broadcasts frequently switch between different groups or perspectives as race dynamics unfold. Unlike long, uninterrupted clips, this segmentation better matches the short focus spans of real broadcasts and the intended application of AR overlays, which only require consistent tracking while a group is in view.

Annotations were created semi-automatically using CVAT (Sekachev et al., 2020), starting with tracking outputs from a baseline tracker and then performing manual corrections to remove false positives, recover missed detections, fix bounding boxes, and resolve identity switches. Although this semi-automatic approach reduces the manual annotation burden, it remains labor-intensive and time-consuming due to the need for frequent corrections, which aligns with observations from prior annotation studies (Scott et al., 2024). Annotations follow the MOTChallenge format (Leal-Taixé et al., 2015) for compatibility with validation tools. Unused fields in 2D evaluation (world coordinates) are filled with default values as required by the format.

A summary of the sequences is illustrated in Fig. 4. Detailed per-sequence statistics are provided in Appendix A. The complete dataset comprises 275,404 annotated bounding boxes across 7007 frames.

4. Methodology

Fig. 5 presents an overview of our proposed pipeline for team recognition and AR visualization in cycling race footage. The system begins with per-frame cyclist detection using a fine-tuned object detection model. Subsequently, a multi-object tracker assigns persistent IDs and maintains identity continuity across frames, even under temporary occlusions or rapid camera movements. This temporal continuity supports per-track aggregation and the placement of AR. It also enables semi-automatic annotation: metadata assigned once per tracked cyclist can be propagated across its entire track.

Team recognition runs one-shot at inference: a Siamese encoder (trained offline and kept frozen) embeds each detected jersey crop and performs distance-weighted matching against an anchor set (three anchors per team), with scores aggregated per team to assign the label. This allows unseen teams to be added by inserting a few anchors, with no fine-tuning or per-race retraining. In parallel, TokenHMR is applied to recover 3D keypoints per rider; we derive a local, rider-centric 3D anchor (an oriented 3D box) from these estimates, ensuring that overlays remain dynamically and accurately positioned without relying on fixed field landmarks or homography. Finally, the AR visualization stage dynamically places text overlays, such as rider names and speeds, based on the 3D bounding boxes. These elements adjust according to cyclists' movements, providing viewers with dynamic information. Each pipeline component is detailed below; associated experiments and results are reported in Section 5. For clarity, the *Temporal Postprocessing* block in Fig. 5 denotes simple per-track smoothing used within AR placement; it is shown separately in the diagram but is not evaluated as a standalone module in this work.

[1] https://universe.roboflow.com/ugent-hkozj/ventoux-chxhg.

[2] Available on Zenodo (DOI: 10.5281/zenodo.17100024).

Fig. 4. Representative frames from the *CyclingTrack* dataset. Top: training sequences. Bottom: test sequences.

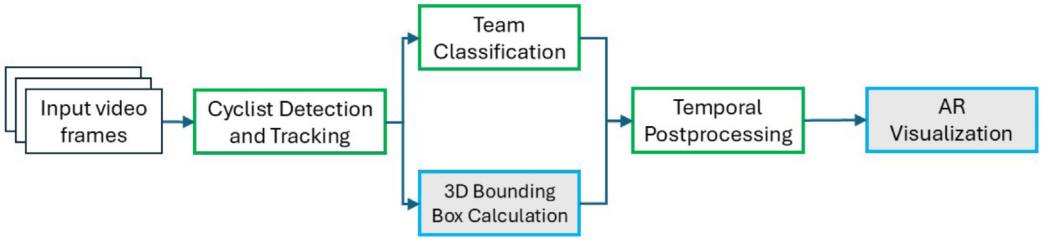


Fig. 5. Overview of the proposed pipeline for cyclist detection, tracking, team recognition, and AR visualization.

4.1. Cyclist detection

Rationale. We compare CNN-based (YOLOv8 (Jocher et al., 2023) and YOLOv11 (Jocher and Qiu, 2024)), and transformer-based (RF-DETR (Robinson et al., 2025)) detectors to assess whether improvements observed on the COCO (Common Objects in Context) dataset (Lin et al., 2015) translate to our broadcast cycling domain. Recent work demonstrates that advances in COCO mAP do not necessarily transfer to more challenging, real-world scenarios with small, dense objects and dynamic backgrounds (Robicheaux et al., 2025). Including RF-DETR enables a fair comparison of state-of-the-art transformers against established CNNs under identical, domain-specific conditions.

Training protocol. All models are fine-tuned on our cyclist detection dataset (see Section 3.1) using single-model, single-size training per experiment. For fair comparison, YOLO and RF-DETR are both trained and evaluated at two input resolutions: 672 × 672 (referred to as *S* for small) and 1792 × 1120 (referred to as *L* for large), which match the stride and divisibility requirements of RF-DETR and enable direct comparison. For each framework, all model scales (YOLO: nano, small, medium, large, and extra-large; RF-DETR: nano, small, medium, and large) are tested. Training uses early stopping (patience: 15 epochs) and identical data augmentation techniques across all models to improve robustness.

Currently, Roboflow's RF-DETR training pipeline does not include data augmentation by default. To ensure parity, we applied an augmentation scheme in RF-DETR that mirrors Ultralytics' YOLO, implemented via the Albumentations library (Buslaev et al., 2020). This includes hue, saturation, and brightness jitter; horizontal flip; random rotation, scaling, translation, and shear; perspective transforms; and mosaic augmentation. Training hyperparameters follow each framework's defaults.

Evaluation protocol. Models are evaluated using the pycocotools implementation of mAP@0.5, with *maxDets* set to 300 to prevent recall saturation in dense peloton scenes (COCO's default of 100 would undercount cyclists in frames with more than 100 detections). We explicitly choose pycocotools for fairness because prior work reported systematic differences between Ultralytics' built-in validator and pycocotools (inflated mAP under the former) (Robicheaux et al., 2025). Evaluation settings are kept identical for all model families and sizes.

Both detection accuracy (mAP@0.5) and inference time are reported from half-precision (FP16) inference outputs, measured on an NVIDIA GeForce RTX 4090 GPU, batch size 1, and including pre- and post-processing steps, to reflect realistic real-time deployment. FP16, or 16-bit floating-point arithmetic, substantially reduces load and memory usage during inference with negligible loss in accuracy (see Appendix B) for comparison with FP32 results. Note that all training is performed in standard 32-bit precision.

4.2. One-shot team recognition

Rationale. The team recognition module is designed to identify cyclists' teams based on the visual appearance of their jerseys, even when encountering new or previously unseen teams. This capability is essential because jersey designs often change throughout the season, such as when riders wear national champion outfits or special editions. It is not practical to retrain a full classification model each time a design changes; therefore, a scalable solution that generalizes across seasons and variations is necessary. To address this, we employ a Siamese neural network approach, which is effective for one-shot learning. In this setup, images are processed through a shared encoder to generate feature embeddings. The network is trained to minimize the distance between embeddings of jersey images for the same team while maximizing the distance for different teams.

Architecture. The Siamese network, shown in Fig. 6, consists of two identical encoders that produce 256-dimensional embeddings, based on the design by Koch (2015). This structure includes five convolutional layers with ReLU activations and 2 × 2 max-pooling, followed by two fully connected layers. We reduced the final embedding size from 4096 to 256 dimensions to lower computational costs and improve recognition efficiency during inference.

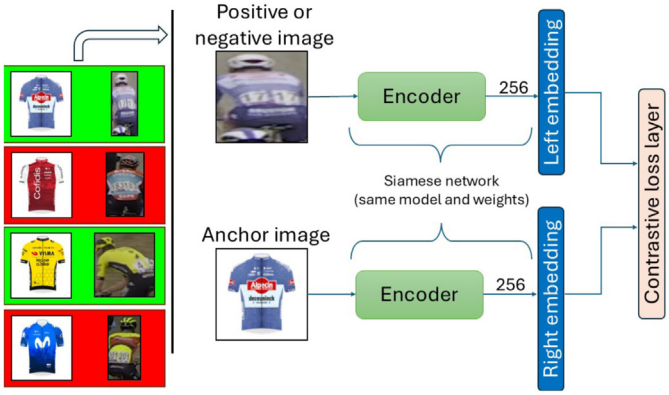


Fig. 6. Siamese network architecture and examples of positive (green border) and negative (red border) pairs used in training.

Using the contrastive loss formula by Pei-Xia et al. (2016), the network is trained to minimize the distance between embeddings of positive pairs (same team) and ensure negative pairs (different teams) are separated by a margin *m*. The loss is defined as:

$$\mathcal{L}(y, d) = (1 - y)\frac{1}{2}d^2 + y\frac{1}{2}\max(0, m - d)^2 \quad (1)$$

where $y \in \{0, 1\}$ indicates whether a pair is dissimilar (1) or similar (0), d is the Euclidean distance between embeddings, and m is the margin.

To support interpretability and provide a binary similarity score during training, a dense layer is added on top of the computed distance to predict whether a given pair represents the same team or not. This allows for a direct performance metric alongside the embedding space learning.

Model usage and one-shot capability. At inference, the Siamese encoder remains frozen. Each detected jersey crop is embedded and compared to all anchors in an anchor set using cosine distance. We use three anchors per team (collected from broadcast footage), and the anchor set is updated per race. We then apply a distance-weighted nearest-anchor

aggregation to obtain per-team scores, enabling easy integration of new teams by adding a few anchors, without any fine-tuning.

To compute confidence scores, we convert inverse distances into a softmax over all anchors:

$$s_i = \frac{e^{d_i^{-1}}}{\sum_{j=1}^{N} e^{d_j^{-1}}} \tag{2}$$

where d_i is the cosine distance between the input crop and anchor i, s_i is the anchor's normalized confidence score, and N is the total number of anchors in the anchor set (three per team in our setup). The inverse distance ensures that closer anchors contribute more to the final score.

To obtain a team prediction, the scores of all anchors belonging to the same team T are summed:

$$S_T = \sum_{i \in T} s_i \tag{3}$$

where S_T is the normalized confidence for team T, and the sum is taken over all anchors i that represent team T. The predicted label is the team with the highest S_T. This method improves interpretability and accuracy, especially when combined with race context. For example, broadcasters typically know which riders or teams are being filmed; by restricting recognition to those specific teams, we reduce the likelihood of misidentifications.

4.3. Cyclist tracking

Reliable tracking of cyclists across frames is essential for per-track aggregation and temporally stable AR overlays. However, tracking cyclists in dynamic, high-speed, and uncontrolled environments, such as road cycling, poses unique challenges compared to traditional sports tracking scenarios due to a large number of athletes, frequent occlusions from environmental elements (e.g., buildings, trees), and the variety of camera angles. We therefore evaluate tracking methods on the *CyclingTrack* benchmark (Section 3.3).

To evaluate tracking performance, we benchmarked state-of-the-art trackers using the BoxMOT framework (Broström), which provides unified configuration, execution, and result logging. The training set was used for hyperparameter tuning via grid search, as default parameters often vary significantly across different application domains, thereby optimizing each tracker for cycling broadcast scenarios. For example, the Intersection over Union (IoU) threshold for detection association was varied to better cope with motion blur and abrupt scale changes. Where applicable, we report variants with and without ReID and specify the ReID backbone.

Four metrics evaluated tracking performance: Higher Order Tracking Accuracy (HOTA) (Luiten et al., 2020), IDF1 (Identity F1) (Ristani et al., 2016), Multi-Object Tracking Accuracy (Bernardin and Stiefelhagen, 2008), and ID switches (IDSW). HOTA was selected as the primary metric due to its balanced focus on both detection quality and association accuracy, capturing both spatial alignment and temporal consistency by evaluating track quality across a range of IoU thresholds. This metric is particularly suitable for broadcasting scenarios where visual overlays must remain stably attached to individual cyclists over time. Compared to MOTA, which is sensitive to detection errors and penalizes identity switches only once, and IDF1, which emphasizes identity consistency but neglects detection quality, HOTA provides a more comprehensive and informative evaluation (Luiten, 2021). All results were computed using the TrackEval toolkit (Jonathon Luiten, 2020). Inference speed was measured by averaging the end-to-end tracking runtime (excluding detection) across all frames and sequences on an NVIDIA GeForce RTX 4090 GPU.

Because *CyclingTrack* deliberately covers diverse broadcast scenarios of different duration, we compute metrics per sequence and report their unweighted mean across sequences (macro-average). This prevents a single long clip from dominating the summary and gives equal weight to each scenario, which aligns with our broadcast use case where short, heterogeneous shots are common.

4.4. 3D bounding box calculation

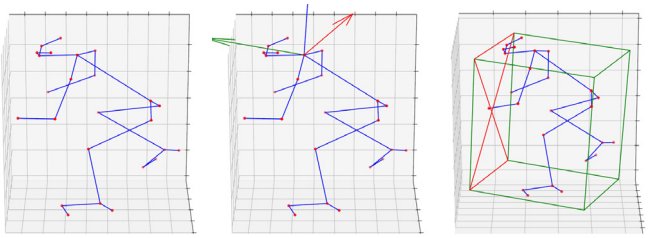
While robust tracking preserves cyclist identities, accurate AR overlays additionally require per-rider 3D geometry. For in-scene AR, traditional 2D bounding boxes are often insufficient to maintain stable alignment with each cyclist's posture as the camera or riders move. We propose a 3D bounding box method based on Human Mesh Recovery using TokenHMR (Dwivedi et al., 2024), which estimates 3D keypoints and an oriented body mesh from a single image crop. This approach helps AR overlays remain visually aligned with each cyclist's posture during camera and rider motion.

Although some mesh recovery models can tolerate partial crops, TokenHMR achieves best results when provided with a full-body input. This presents a practical challenge, as our cyclist detector is trained specifically to localize the upper body (maximizing jersey visibility for team recognition), rather than the entire cyclist. As a result, an additional step is required to transform upper-body detections into suitable full-body crops for robust mesh estimation.

In the previous version of our work (Clinckemaillie et al., 2026), this was addressed using a two-stage detection pipeline. First, our upper-body YOLO (Jocher and Qiu, 2024) detector localized the torso and head region of each cyclist. Second, a generic person detector was applied to the same frame to detect full-body bounding boxes. Detections from both stages were associated at IoU ≥ 0.3 to retain cyclist-specific full bodies. This approach ensured that TokenHMR received a complete input. However, it had notable drawbacks: the generic person detector could fail to detect cyclists, resulting in missed crops and interrupted tracking. Additionally, the requirement of a second detector added computational overhead and pipeline complexity.

In this paper, we streamline the process to improve both speed and robustness in practical use. Instead of relying on a separate person detector, we use only the upper-body detector and expand each detected bounding box by a fixed, empirically determined padding factor to approximate a full-body crop. This approach is especially effective in sparse peloton scenarios, where occlusion is limited, and provides sufficiently stable crops for mesh recovery. If group density increases, the previous two-stage detection method can be re-enabled as a fallback without further pipeline changes.

Fig. 7 illustrates the procedure for calculating a cyclist's 3D bounding box. Keypoints on the cyclist are first extracted (see Fig. 7(a)) to define orientation axes. The x-axis is drawn from the left to the right hip, the y-axis is the average of vectors from the hips to the hands and along the spine, and the z-axis is the cross product of the x- and y-axes. The y-axis is then recalculated using the new z- and x-axes to maintain orthogonality, as illustrated in Fig. 7(b). Keypoints are projected onto these axes to find extreme values, from which the bounding box corners are derived (see Fig. 7(c)). This results in 3D bounding boxes that account for each cyclist's translation and rotation for consistent projection onto the original 2D frame.



(a) Keypoints extraction (b) Estimated orientation (c) Bounding box corners

Fig. 7. Steps for calculating a 3D-oriented bounding box.

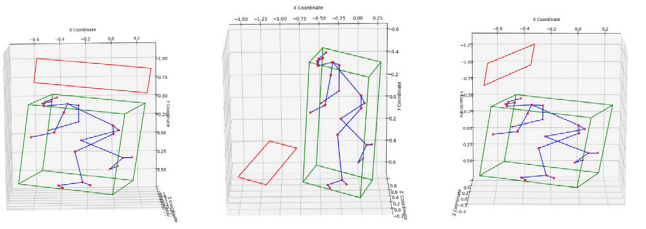
4.5. AR visualization

This section describes how 3D geometry from mesh recovery enables robust integration of AR elements into cycling broadcast footage.

We use 3D bounding boxes to position virtual overlays, such as rider names and race statistics, around individual cyclists. By focusing on smaller groups, such as breakaways, we avoid visual overload, as applying AR to the entire peloton could become confusing and quickly exceed current technical limits. Breakaway groups are also easier for team recognition, because they usually include riders from different teams, and broadcasters often know in advance which riders are in the group.

Our system uses this group information to restrict team recognition to the actual riders present in the filmed group. This allows us to generate overlays that display team and rider-level information for each cyclist in the breakaway, instead of only general team labels. Since the group composition is known, detected teams can be linked to specific riders, resulting in more informative and targeted AR visualizations. A limitation of this approach is that it does not scale to full-peloton helicopter shots. When multiple riders from the same team are visible, team recognition alone is insufficient to resolve individual identities, especially at small scales and under occlusion.

AR elements are positioned using the 3D bounding boxes described in Section 4.4. The size and location of these overlays are determined by the dimensions and orientation of the bounding box. The text area's height matches the bounding box height, while its width is set by the aspect ratio of the overlay content. There are multiple ways to position the text area, as illustrated in Fig. 8. Three examples of possible placements include: above the rider, aligned with their body orientation (Fig. 8(a)); on the ground ahead of the rider (Fig. 8(b)); and above the rider but oriented perpendicular to their direction (Fig. 8(c)). In this study, the placement variant is selected manually; automatic selection is out of scope.



(a) Positioned above the rider (b) On the ground ahead of the rider (c) Perpendicular to the rider's path

Fig. 8. Several options for positioning the text area.

Text overlays are projected onto the original image by mapping 3D corner points to 2D image coordinates using OpenCV (Bradski, 2000), as illustrated in Fig. 9. Alpha compositing is utilized to make the text semi-transparent; however, blending operations may introduce processing overhead. Optimized GPU support helps mitigate some of these extra costs.

Improvements are made to stabilize overlays and reduce the chaotic visualizations caused by noisy bounding box positions and orientations across frames. First, for each frame, we compute a mean orientation axis across all detected cyclists rather than estimating orientation per rider. This enforces a uniform text orientation within the frame and yields a more stable view. Fig. 10(a) illustrates the per-rider orientation baseline, while Fig. 10(b) shows the uniform rotation obtained by averaging the orientation axes.

(a) Text placement using individual cyclist's orientation. (b) Text placement using averaged orientation.

Fig. 10. Comparison of text placement strategies.

Second, we temporally smooth the text-area corner points using tracked data from previous frames. We apply exponential smoothing to the 2D corner coordinates and a short-window moving average to the orientation axes, which reduces jitter and produces smoother motion.

A demo showcasing both 3D-aligned text and static name overlays is available as supplementary material at: https://youtu.be/0BbghrJUJmM. A user-centered evaluation of readability, informativeness, visual calmness, and engagement is reported in Section 6, where we compare the proposed 3D overlays against conventional labels.

5. Experiments & results

5.1. Cyclist detection

We report results on the held-out test split of the cyclist detection dataset, using the training and evaluation protocols from Section 4.1. Table 1 summarizes mAP@0.5 and inference time for all YOLOv8 (Jocher et al., 2023), YOLOv11 (Jocher and Qiu, 2024), and RF-DETR (Robinson et al., 2025) variants on the test set (all results in FP16). For a comparison with FP32, see Appendix B.

The results show that RF-DETR-Large outperforms all other models in detection accuracy at both resolutions, achieving the highest mAP values (95.5 at large, 86.1 at small). However, this comes at a significant computational cost: its inference time is substantially higher than any of the YOLO variants, making it impractical for real-time deployment in our pipeline.

When considering models suitable for real-time use, YOLOv11 and RF-DETR achieve very similar performance, with YOLOv11 models generally providing faster inference. YOLOv8 consistently lags behind both YOLOv11 and RF-DETR, especially at high resolution. It is also notable that the performance difference between model sizes is relatively modest for this dataset (except at the Nano scale, where accuracy drops off more quickly). This pattern holds for both model families, suggesting that increasing model size alone does not yield substantial

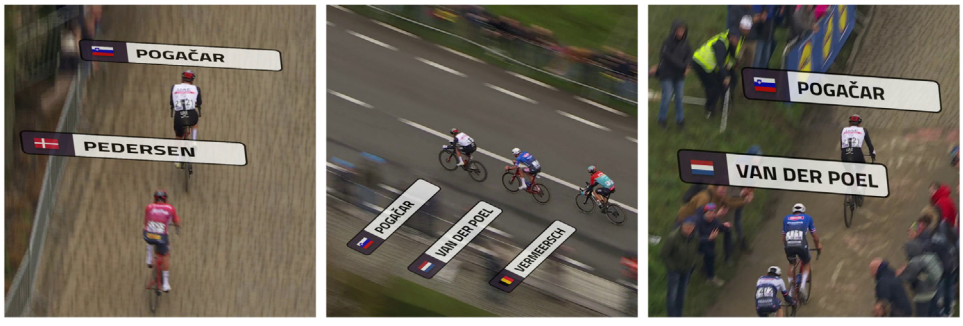


Fig. 9. Examples of AR visualizations.

Table 1
mAP@0.5 (in percentage) and inference time (in ms) for YOLOv8 (Jocher et al., 2023), YOLOv11 (Jocher and Qiu, 2024), and RF-DETR (Robinson et al., 2025) across model scales at two input resolutions: large (1792 × 1120) and small (672 × 672) on the test set. Best mAP per resolution and fastest time in bold; second-best underlined.

Model	mAP@0.5		Time (ms)	
	L	*S*	*L*	*S*
YOLOv8-Nano	87.8	68.3	**7.3**	**4.1**
YOLOv8-Small	88.9	76.7	8.7	4.7
YOLOv8-Medium	88.9	77.6	12.3	6.0
YOLOv8-Large	89.5	78.0	15.3	6.9
YOLOv8-XLarge	87.7	78.7	23.1	9.2
YOLOv11-Nano	90.3	70.4	7.7	4.9
YOLOv11-Small	92.7	76.2	9.1	5.0
YOLOv11-Medium	92.7	78.2	12.2	5.9
YOLOv11-Large	92.8	79.6	14.3	7.0
YOLOv11-XLarge	92.8	77.5	21.1	9.0
RF-DETR-Nano	90.0	67.6	17.8	4.6
RF-DETR-Small	92.1	70.9	19.7	4.9
RF-DETR-Medium	92.7	68.6	18.1	4.9
RF-DETR-Large	**95.5**	**86.1**	41.6	9.8

gains for the cyclist detection task, likely due to the dominance of small objects in dense broadcast footage.

Based on these findings, YOLOv11-Large at high resolution is the most practical choice. It offers a good trade-off between detection accuracy (92.8 mAP) and inference speed (14.3 ms), which is essential for reliable tracking and team recognition. In situations where additional downstream modules, such as 3D pose estimation or full AR rendering, increase the computational load, smaller models like YOLOv11-Small or Nano can provide further speed-ups (down to 7.7 ms per frame) with only a moderate decrease in accuracy. This flexibility allows the pipeline to be adapted to different system requirements and hardware constraints, balancing speed and accuracy as needed.

5.2. One-shot team recognition

We evaluate the Siamese encoder described in Section 4.2 on the jersey-crop dataset summarized in Section 3.2. To evaluate pairwise classification, we utilized the output from the dense layer. As a baseline configuration, each team was represented by only one online-sourced anchor image (as shown in Fig. 6), and negative sampling was performed randomly, which resulted in a validation accuracy of 71%.

We then enhanced our approach by replacing the single anchor image with three images taken from actual race footage, each captured from a distinct viewpoint (e.g., front, side, above). This modification aimed to create a more comprehensive similarity model that better reflects how jerseys might appear in real conditions, rather than relying on a single reference that may not generalize well. Furthermore, using three anchors for each team allowed us to generate multiple positive (one for each anchor) and negative pairs (from other teams) for each training sample. Additionally, we employed hard negative mining to target visually similar jerseys by explicitly sampling negative examples from teams with nearly identical color schemes. By intentionally exposing the model to these challenging cases, we reduced the number of false positives associated with near-identical apparel. We empirically selected the margin hyperparameter used in the contrastive loss by conducting a grid search over values from 0.5 to 1.5 (with step size 0.2). The best validation performance was achieved at $m = 0.7$, which was therefore used throughout the experiments. Overall, these refinements increased the dense layer's validation accuracy to 83%, demonstrating a significant improvement in our ability to distinguish images of the same team from those of different teams.

We report overall accuracy (micro) together with macro per-team recall (i.e., class-wise accuracy) to capture variation across teams. On the validation set, overall accuracy was 94%, and the macro per-team recall was 95% ± 12% (minimum 67%), confirming its effectiveness for seen jerseys during training. More importantly, on the test split with held-out teams (classes not used for training; three labeled anchors per team at inference; no fine-tuning), overall accuracy was 84% and macro per-team recall was 85% ± 12% (minimum 60%), demonstrating good one-shot generalization to unseen teams. Errors are concentrated in small-crop cases and in confusion between visually similar jerseys; we quantify the role of input crop sizes below.

To further analyze the model's performance, we examined the influence of input crop sizes on classification accuracy and confidence. Crops were grouped by their largest side length, and both classification accuracy and average confidence were calculated per size category. As shown in Fig. 11, most crops in the validation set fall between 50 and 100 pixels. Table 2 summarizes the results per group. Accuracy consistently increased with image size, reaching 100% for the 125–150 and >175 pixel categories. Smaller images (<50 pixels) had the lowest accuracy (83%) and confidence (0.19), suggesting that visual detail is a key factor for reliable classification. The average confidence score followed a similar trend, growing steadily with crop size and reaching 0.35 for the largest inputs. In practice, we only provide a team prediction when the largest side of the detected crop is at least 50 pixels, as smaller inputs are too unreliable.

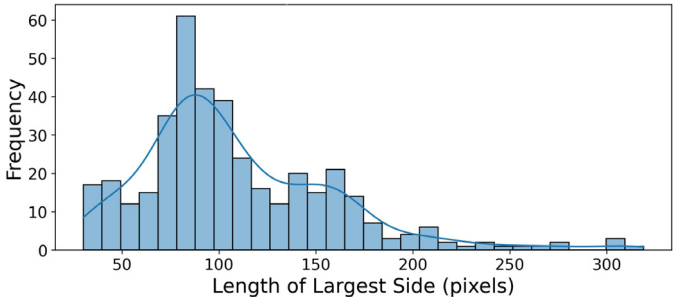


Fig. 11. Distribution of largest side lengths of crops in the validation set.

Table 2
Classification accuracy and average confidence per image size category (largest side, in pixels).

Image size (px)	Accuracy	Average confidence
<50	0.83	0.19
50–75	0.90	0.22
75–100	0.92	0.23
100–125	0.97	0.25
125–150	1.00	0.31
150–175	0.98	0.27
>175	1.00	0.35

Although small-crop scenarios pose challenges, we can partially address this by accumulating evidence over time through temporal tracking across consecutive frames (e.g., a short moving average of confidence scores). Overall, these results highlight the effectiveness of Siamese networks for one-shot team recognition and underscore the importance of resolution. Crucially, this framework enables quick adaptation to newly introduced teams, minimizing the need for extensive retraining whenever a new jersey design appears.

5.3. Cyclist tracking

To isolate tracking performance from detection quality, all trackers were evaluated using the same set of detections; the per-frame bounding boxes produced by the YOLOv11-large detector (see Section 5.1). Re-identification features were precomputed using either the pretrained OSNet model (Zhou et al., 2019) or the custom encoder trained for one-shot team recognition (see Section 5.2), which we refer to as CyclingNet. Although not explicitly designed for re-identification, CyclingNet substantially reduces embedding extraction time to 2.9 ms per frame, compared to 14.4 ms per frame for OSNet on our dataset. Tracking results on all sequences are presented in Table 3.

Overall, a clear trade-off is observed between tracking accuracy and inference speed. StrongSORT (Du et al., 2023), especially when combined with the CyclingNet encoder, achieves the highest HOTA score (0.72), but runs at only 27 fps, which is insufficient for real-time use when combined with other system components (such as cyclist detection). In contrast, Deep OC-SORT (Maggiolino et al., 2023) achieves competitive performance (0.69 HOTA), while running over 800 fps, offering substantial headroom for other modules like team classification and AR rendering. Notably, BoT-SORT-ReID (Aharon et al., 2022) also delivers comparable accuracy (0.69 HOTA) at 87 fps, balancing robustness and speed.

The use of our CyclingNet consistently improves performance across all Re-ID-capable trackers compared to the OSNet baseline, despite the latter being pretrained for person re-identification. This suggests that CyclingNet captures highly discriminative jersey features specific to the cycling domain. Its lightweight nature also enables more efficient pipelines and reduces reliance on heavyweight Re-ID models.

We further analyze sequence-level performance in Fig. 12, which visualizes per-sequence HOTA (top; represented in percentage) and frames per second (bottom; log scale). The red line at 50 fps marks our real-time feasibility threshold, accounting for the full pipeline operating at 25 fps. Results show that per-sequence HOTA and inference time are heavily influenced by scene characteristics such as cyclist density and viewpoint.

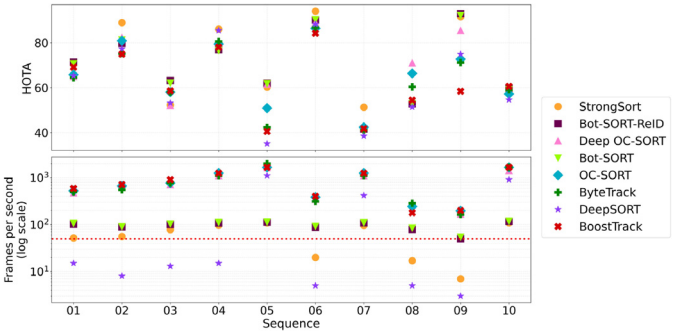


Fig. 12. Per-sequence tracking performance on the *CyclingTrack* dataset. Top: HOTA score (in percent). Bottom: inference speed in frames per second (log scale). The red horizontal line in the bottom plot indicates the 50 fps threshold for real-time feasibility.

StrongSORT consistently excels in occlusion-heavy scenarios like *Seq02* and *Seq07*, where long-term re-identification is crucial but challenging due to frequent disappearances behind trees or buildings. However, globally, dense peloton shots with wide or top-down views (such as *Seq06*, *Seq08*, and *Seq09*) reveal a significantly narrowed performance gap between StrongSORT and faster alternatives like Deep OC-SORT and BoT-SORT-ReID. These shots typically involve helicopter footage, with fewer occlusions and a more consistent appearance across frames, which favors simpler association mechanisms.

This motivates a hybrid tracking strategy: StrongSORT could be selectively enabled in sparse scenes or high-occlusion conditions, while Deep OC-SORT can serve as a fast fallback for dense, stable peloton views. Although dynamic switching is non-trivial, a lightweight scene classification module with hysteresis could support adaptive strategies.

It is also noteworthy that sequences involving low-angle front views with heavy occlusions (*Seq05* and *Seq07*) consistently yield the lowest HOTA scores, confirming that such viewpoints pose fundamental challenges to multi-object tracking. The combination of occlusion, motion blur, and rapid size changes likely breaks the temporal coherence needed for reliable ID assignment.

5.4. Inference time and optimization

To evaluate the real-time feasibility of our pipeline for live cycling broadcasts, we benchmarked per-frame inference times on a representative video featuring four cyclists prominently visible in the frame, typical of breakaway scenarios. Measurements were taken on an NVIDIA GeForce RTX 4090 GPU, including pre- and post-processing steps. As shown in Fig. 5, modules highlighted in green meet the desired real-time speed of 25 fps, while those in blue represent components that *were* computational bottlenecks prior to optimization. Table 4 summarizes the inference times for each module. Oriented 3D bounding box calculation and per-track smoothing are excluded from the table due to their negligible execution times (less than 1 ms per frame).

Table 4
Per-frame inference time measured on an NVIDIA GeForce RTX 4090 GPU for core vision modules (including pre- and post-processing).

Module	Model/Method	Time per frame (ms)
Cyclist detection	YOLOv11-large	14.3
Tracking	Deep OC-SORT	1.2
Team recognition	CyclingNet	3.9
3D pose estimation	TokenHMR	95.2
Total	–	**114.6**

Initially, the end-to-end pipeline achieved ~8 fps and thus fell short of the 25 fps target. To address this, we applied TensorRT-based acceleration to the two dominant modules: YOLOv11-large and TokenHMR (Corporation, 2025). TensorRT is NVIDIA's SDK and inference runtime that compiles trained neural networks into optimized GPU engines using techniques such as reduced-precision execution, layer/kernel fusion, and kernel auto-tuning to minimize latency and

Table 3
Performance of different trackers on the *CyclingTrack* dataset. CyclingNet denotes our jersey-embedding encoder reused as lightweight Re-ID features (see Section 4.2); OSNet is a pretrained person Re-ID backbone (Zhou et al., 2019). Variants are grouped by tracker; within each tracker, rows are sorted by HOTA. Best per metric in bold, second-best underlined. Lower is better for IDSW. FPS excludes detection time.

Tracker	Re-ID	HOTA (↓)	IDF1	MOTA	IDSW	FPS
StrongSORT (Du et al., 2023)	CyclingNet	**0.72**	0.74	0.85	842	27
	OSNet	0.67	0.70	0.83	973	21
BoT-SORT-ReID (Aharon et al., 2022)	CyclingNet	0.69	**0.75**	0.80	691	87
	OSNet	0.68	**0.75**	**0.89**	713	87
Deep OC-SORT (Maggiolino et al., 2023)	CyclingNet	0.69	0.67	0.84	747	802
	OSNet	0.68	0.71	0.84	**660**	819
BoT-SORT (Aharon et al., 2022)	N/A	0.69	**0.75**	0.79	663	91
OC-SORT (Cao et al., 2023)	N/A	0.66	0.69	0.83	869	**861**
ByteTrack (Zhang et al., 2022)	N/A	0.64	0.68	0.82	2300	475
DeepSORT (Wojke et al., 2017)	CyclingNet	0.62	0.64	0.68	981	8
	OSNet	0.60	0.62	0.67	1212	7
BoostTrack (Stanojevic and Todorovic, 2024)	N/A	0.62	0.67	0.86	1432	551

maximal throughput. We use it to build FP16 engines for YOLOv11 and TokenHMR, which underpins the speedups reported below.

Cyclist detection optimization. For the YOLOv11-large model, utilizing TensorRT with FP16 precision significantly reduced the inference time from 14.3 ms to 8.1 ms per frame. Importantly, the detection accuracy remained stable, with only a minor decrease in mAP@0.5, dropping from 0.937 to 0.928.

Mesh recovery optimization. We compared static and dynamic TensorRT engines for TokenHMR, which resizes inputs to 256 × 256 pixels, differing only in batch size. Dynamic engines offer flexibility for varying batch sizes but come with computational overhead. As illustrated in Table 5, static engines are significantly faster, especially for smaller batch sizes. We therefore opted for the static engine at a batch size of 4 (17.2 ms). This limits processing to four cyclists per pass and is typically sufficient for on-air overlays. For smaller groups of cyclists, dummy inputs can be used to pad the batch, while larger groups can be managed through selective visualization. This approach ensures optimal inference speed and efficient resource utilization.

Table 5
Comparison of static versus dynamic TensorRT engines for TokenHMR across varying batch sizes.

Batch size	Static engine (ms)	Dynamic engine (ms)
1	6.9	17.2
2	11.2	18.0
4	17.2	23.2
6	25.6	29.3
8	31.6	33.0
12	49.3	55.0
16	63.2	66.3
32	134.8	135.9

With these optimizations, the measured vision stack drops from 114.6 ms to 30.4 ms per frame (~33 fps), meeting the 25 fps requirement on dedicated hardware.

In addition to these module-level improvements, deployment-level strategies offer further advantages:

- Horizontal scaling with a fixed one-frame offset: Cyclist detection and tracking run on GPU#1, while TokenHMR operates on GPU#2, using the tracks from the previous frame. This produces HMR results that are one frame behind the live video. If needed, buffering the video by one frame synchronizes HMR with the display, resulting in a consistent one-frame end-to-end delay. Without buffering, only the HMR component is offset; the rest of the pipeline remains live.
- Single-GPU with controlled latency: If a multi-frame latency is acceptable, sparse keyframe processing combined with interpolation can stabilize overlays while reducing TokenHMR computations.
- Asynchronous team recognition: As with team recognition (see Section 5.2), this module can run asynchronously after enough data has been collected for each track, decoupling it from strict per-frame processing.

Conclusion Together, TensorRT optimizations demonstrate 25 fps feasibility for our pipeline on dedicated hardware (approximately 33 fps). On slower hardware, the same target can be achieved by accepting a small, fixed latency (e.g., one frame via dual-GPU scheduling, or a few frames with keyframe processing).

6. User experience evaluation

The previous chapters have demonstrated that our pipeline can integrate dynamic, 3D pose-based AR overlays into live cycling broadcasts. However, technical performance alone does not ensure a better viewing experience. Traditional broadcasts typically rely on static or semi-automatically tracked rider labels, and it remains uncertain whether our proposed overlays provide a tangible advantage over this established practice. To evaluate their practical utility, we conducted a user experience (UX) study that systematically compared different AR strategies against both standard broadcast graphics and unaltered footage. The study focused on readability, informativeness, visual clutter, and viewer engagement, aiming to gain initial insights into audience preferences and the perceived added value of our system in a realistic broadcast context.

6.1. Study design

The user evaluation was conducted through an online survey.[3] In the introductory part of the survey, respondents were asked a small set of demographic questions (age, gender, viewing habits, knowledge of cycling, and previous experience with AR) to enable exploratory analysis of potential associations with perceived AR overlay quality.

Participants were recruited by distributing the Google Forms link via the internal mailing list of our research group, as well as through private messages to a small number of acquaintances. All participants were volunteers, and none had prior involvement in the project or detailed knowledge of the technical aspects under evaluation. While this approach yielded a modest and diverse sample, the primary aim was to gather initial impressions and qualitative feedback rather than achieve statistical generalization.

The main body of the survey consisted of four video fragments, each showing a small breakaway group of riders. This setting was deliberately chosen, as a limited number of athletes on screen reduces visual clutter and allows for a clearer assessment of overlay visibility and perceived distraction. Fig. 13 illustrates the three versions of one representative fragment, while the full set of survey examples across all fragments is provided in Appendix C.

- Version A (Fig. 13(a)), the original broadcast footage without overlays, serving as a baseline.
- Version B (Fig. 13(b)), a 2D-rider label overlay similar to those used in current broadcast productions (output of our pipeline without 3D pose-based alignment).
- Version C (Fig. 13(c)), our proposed dynamic, 3D pose-based AR overlay.

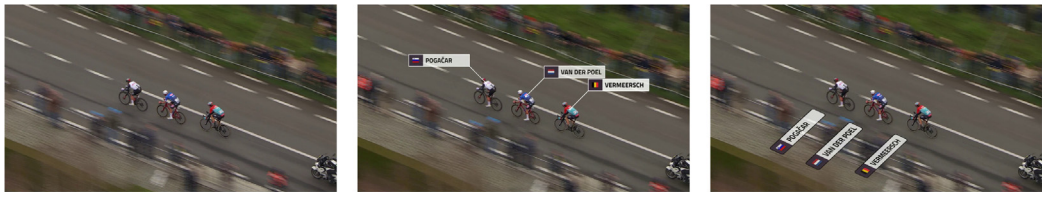
To minimize bias, the order of versions B and C was randomized within the survey; however, for the remainder of this paper, we refer to B as the conventional overlay and C as the proposed AR overlay. Across the four fragments, two implemented the pose-based overlay directly above the riders, while the other two projected the overlay onto the ground plane, enabling a comparison of how spatial placement affects perception.

Table 6
Survey questions used in the UX survey (5-point Likert scale unless noted otherwise).

Variable	Question
For each overlay version	
Readability	The overlay was easy to read.
Informativeness	The overlay provided useful additional information.
Visual calmness	The overlay was visually calm and non-disturbing.
Engagement	The overlay made the clip more engaging to watch.
For each fragment	
Preference	Which version do you prefer? (A/B/C/No preference)

After each overlay (B/C), participants answered four 5-point Likert questions targeting user experience (see Table 6). At the end of each fragment, they also indicated an overall preference (A, B, C, or no

[3] A duplicate of the survey is available here: https://forms.gle/Lzce23ndw56X1Ywu5.



(a) Original footage without overlay (b) Conventional static labels (c) Proposed 3D pose-based AR overlay

Fig. 13. Example survey conditions.

preference) and could provide an optional comment. This yielded complementary quantitative scores and qualitative feedback.

6.2. Survey results

Nineteen participants completed the survey. Although basic demographics (age, viewing frequency, cycling knowledge, AR experience) were collected to enable exploratory analyses, no clear associations with the questionnaire outcomes emerged in this small sample. Given the small sample size, we limit our analysis to a descriptive summary of the main user responses.

Fig. 14 summarizes the forced-choice preference per fragment and overall. Across fragments, the conventional overlay (B) was preferred by the majority of respondents. Aggregated over all fragments, 79% selected B, 10% selected C, and 10% selected the baseline video without overlays (A). This pattern suggests that, in its current form, the conventional visualization remains the safer default.

Fig. 15 shows aggregated Likert ratings (per respondent means) for readability, informativeness, visual calmness, and engagement; per-fragment distributions are provided in Appendix D (Fig. D.17). Below, we summarize each item with representative themes from the optional comments.

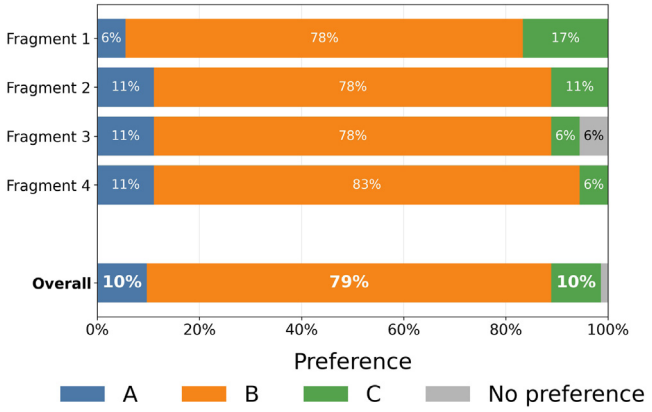


Fig. 14. Per-fragment and overall overlay preference. A: no overlay, B: conventional static labels, C: pose-based AR.

Readability. Conventional overlays received higher median readability scores and showed less variability among respondents than pose-based labels. Per-fragment plots Appendix D indicate that the difference was most pronounced in Fragment 2, where pose-based labels were projected on the ground plane with a pronounced leftward rotation relative to the riding direction. Several participants described this as "harder to read" or "unnatural".

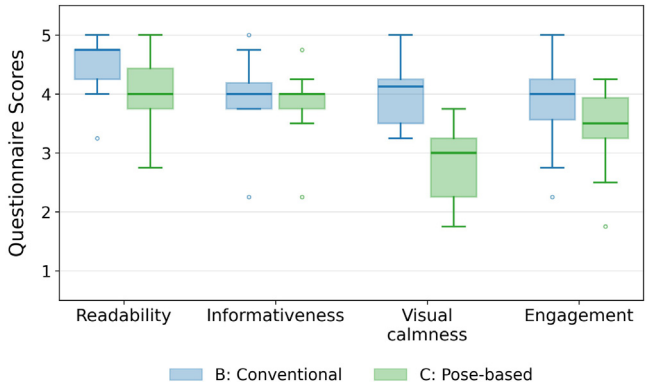


Fig. 15. Boxplot with aggregated Likert scale ratings (1–5) for readability, informativeness, visual calmness, and engagement.

Informativeness. Both overlays scored similarly (around the "agree" range), consistent with the fact that they conveyed the same identity information. Comments indicated that name labels are most useful when identity is ambiguous (e.g., dense groups or less prominent riders), and less necessary when riders are widely recognizable. A few respondents noted that connector lines or pointers (overlay B) help link labels to riders when not every rider is annotated.

Visual calmness. Pose-based overlays showed consistently lower medians and a wider spread. Participants frequently cited overlap between labels, fast relative motion of text planes, and perspective-driven size/orientation changes as sources of distraction. Participants suggested avoiding label collisions, stabilizing motion, and preventing labels from intersecting riders.

Engagement. Differences between B and C were minor and fragment-dependent. Some participants found C visually interesting in specific shots, but others described it as "unnatural" or "busy", which likely offsets potential gains in engagement.

6.3. Takeaways and implications

Overall preferences and question ratings converge on the same conclusion: in its current state, the pose-based presentation does not yet match the perceived readability and visual calmness of the conventional overlay. The qualitative feedback points to concrete next steps:

(i) robust collision avoidance and occlusion handling to prevent label–label and label–rider overlap; (ii) temporal smoothing and hysteresis to further reduce jitter and sudden re-orientations; (iii) constraints on rotation and perspective distortion, combined with a minimum dwell time, to preserve legibility; and (iv) road-plane placement only, implemented via a road-segmentation model that restricts text planes to road pixels and masks riders/background, thereby avoiding intersection with cyclists and distracting scenery. Beyond identity labels, (v) selectively presenting context-aware metrics (e.g., time gaps, gradient, speed) may improve perceived informativeness and engagement when identity is already obvious, while omitting labels in shots where they add limited value. Finally, even in the conventional overlay setting (B), the system improves on current practice: rather than manually selected riders and unstable pixel-based tracking, it provides automated, frame-consistent labels for all visible riders, offering practical value irrespective of the chosen AR style.

7. Limitations

While the system achieves stable, team-aware AR overlays on helicopter footage, several limitations remain:

- Team recognition degrades for very small crops (largest side <50 px) and for visually similar jerseys. We mitigate this by abstaining from predicting below 50 px; in addition, temporal aggregation over tracks and broadcast context (restricting candidates to known participants) stabilizes predictions.
- Identification is currently at the team level; mapping to individual riders relies on broadcast context (known breakaway composition) and becomes ambiguous when multiple riders of the same team are in view.
- The tracking setup does not yet exploit cycling-specific priors (e.g., group-motion/heading coherence). We also do not evaluate re-identification across extended out-of-view gaps; our broadcast use case focuses on maintaining identity while a group remains in view.
- Overlay layout is not yet fully automatic: collision avoidance and occlusion-aware placement are currently not implemented, and the placement variant (e.g., body-aligned or on road-plane) is configured manually per scene. In our user experience study, most participants preferred conventional broadcast labels in the current prototype, primarily due to label–label overlap and occasional temporal instability (jitter).
- The compositor uses simple alpha blending and does not model illumination; lighting, shadows, and color shifts can affect perceived realism and embedding stability.

8. Conclusion

This work presented an end-to-end vision pipeline that augments live cycling broadcasts with team-aware, pose-aligned AR. The system integrates cyclist detection (YOLOv11), multi-object tracking, one-shot team recognition, and mesh-based 3D pose estimation (TokenHMR) to produce stable overlays on top of helicopter footage.

On our detection dataset, YOLOv11-Large at 1792×1120 achieves 0.93 mAP@0.5 with a mean inference time of 14.3 ms per frame (reduced to 8.1 ms using TensorRT/FP16), providing reliable inputs for downstream modules. On the *CyclingTrack* benchmark, a hybrid strategy emerges: appearance-aware association (StrongSORT) excels in sparse, occlusion-heavy shots, whereas lightweight trackers (e.g., (Deep) OC-SORT, BoT-SORT-ReID) offer strong accuracy-speed trade-offs in dense, top-down peloton views. For team recognition, a Siamese encoder with three anchors per team generalizes to unseen teams without fine-tuning, achieving an 85% mean accuracy. Temporal aggregation over tracks and simple broadcast priors further increases robustness. AR overlays are positioned using per-rider 3D geometry from TokenHMR. We derived an oriented 3D box from the recovered mesh and projected its corners via a perspective transform; overlays were then composited with alpha blending. Lightweight temporal smoothing (axis averaging and exponential filters) reduced jitter and kept overlays coherent under camera motion. With TensorRT optimizations, the whole vision stack runs at ~30 ms per frame (~33 fps), demonstrating 25 fps feasibility on dedicated hardware.

Remaining limitations are summarized in Section 7. Future work could address these limitations by enabling rider-level identification by integrating sensor data (bike positions), automating overlay placement with road segmentation plus collision/occlusion handling, and improving visual realism by modeling lighting and shadows.

In conclusion, this work demonstrates the potential of integrating computer vision and AR technologies to transform the viewing experience of cycling races. By advancing cyclist detection, tracking, team recognition, and AR visualization, this research lays the groundwork for more engaging and informative sports broadcasts. Continued exploration and refinement of these technologies will be crucial in addressing current limitations and expanding their applicability, ultimately enhancing the immersive viewing experience for cycling enthusiasts.

CRediT authorship contribution statement

Winter Clinckemaillie: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Jelle Vanhaeverbeke:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Maarten Slembrouck:** Writing – review & editing, Supervision. **Steven Verstockt:** Writing – review & editing, Supervision, Resources, Funding acquisition.

Funding

This research was funded by imec and the Flemish Government's Department of Culture, Youth and Media within the project called Digital Transformation Media, grant number 94186.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the use of Grammarly for AI-assisted proofreading of this manuscript.

Appendix A. Cyclingtrack sequence overview

See Table A.7.

Appendix B. FP32 vs. FP16 cyclist detection comparison

To assess the effect of numerical precision, Table A.8 shows detection results for all detection models in full precision (FP32), using the same test split and evaluation protocol as described in Section 4.1.

Table A.7
Overview of the sequences included in the *CyclingTrack* dataset.

Training sequences					
Name	Length	Tracks	Boxes	Density	Scene description
Seq01	250 (00:10)	78	11,070	44.64	Low-angle rear view; dense peloton with frequent occlusions
Seq02	501 (00:20)	38	15,846	31.75	High-angle side view; sparse peloton, occlusions from trees
Seq03	897 (00:36)	71	18,269	28.28	Low-angle front view; dense peloton with heavy occlusions
Seq04	2125 (01:25)	12	24,936	11.76	Varying viewpoints; sparse peloton, environmental occlusions
Seq05	395 (00:16)	6	1518	3.87	Low-angle side view; sparse peloton, occlusions from spectators
Total training	**4168 (02:47)**	**205**	**71,639**	**17.18**	
Testing sequences					
Name	Length	Tracks	Boxes	Density	Scene description
Seq06	582 (00:24)	73	37,488	64.85	Top-down view; dense peloton, occlusion from trees
Seq07	678 (00:27)	16	8157	12.06	High-angle side view; sparse peloton, long building occlusions
Seq08	156 (00:07)	150	10,765	69.90	Bird's-eye frontal view; fast, small cyclists with long shadows
Seq09	1125 (00:45)	128	145,640	129.92	High-angle overview; very dense peloton, small cyclists
Seq10	298 (00:12)	22	1715	5.80	Low-angle side view; fast motion, strong occlusions
Total testing	**2839 (01:55)**	**359**	**203,765**	**71.79**	

Table A.8
mAP@0.5 (in percentage) and inference time (in ms) for YOLOv8 (Jocher et al., 2023), YOLOv11 (Jocher and Qiu, 2024), and RF-DETR (Robinson et al., 2025) across model scales at two input resolutions: large (1792 × 1120) and small (672 × 672) on the test set in FP32. The value in parentheses indicate the difference compared to FP16 results in Table 1 (red: slower or lower mAP; green: faster or higher mAP). Best mAP per resolution and fastest time in bold; second-best underlined.

Model	mAP@0.5		Time (ms)	
	L	*S*	*L*	*S*
YOLOv8-Nano	88.3 (+0.5)	68.7 (+0.4)	**8.5** (+1.2)	**4.4** (+0.3)
YOLOv8-Small	89.6 (+0.7)	76.8 (+0.1)	10.7 (+2.0)	5.6 (+0.9)
YOLOv8-Medium	89.1 (+0.2)	77.3 (−0.3)	18.7 (+6.4)	7.4 (+1.4)
YOLOv8-Large	89.9 (+0.4)	78.7 (+0.7)	24.9 (+9.6)	8.9 (+2.0)
YOLOv8-XLarge	88.5 (+0.8)	78.0 (−0.7)	39.9 (+16.8)	12.9 (+3.7)
YOLOv11-Nano	91.0 (+0.7)	70.5 (+0.1)	8.7 (+1.0)	5.1 (+0.2)
YOLOv11-Small	93.3 (+0.6)	76.7 (+0.5)	10.9 (+1.8)	5.1 (+0.1)
YOLOv11-Medium	93.1 (+0.4)	78.5 (+0.3)	17.3 (+5.1)	6.9 (+1.0)
YOLOv11-Large	93.7 (+0.9)	79.6 (–)	20.8 (+6.5)	8.0 (+1.0)
YOLOv11-XLarge	93.7 (+0.9)	77.9 (+0.4)	36.1 (+15.0)	11.5 (+2.5)
RF-DETR-Nano	90.0 (–)	67.5 (−0.1)	41.8 (+24.0)	7.9 (+3.3)
RF-DETR-Small	92.7 (+0.6)	71.4 (+0.5)	48.9 (+29.2)	7.4 (+2.5)
RF-DETR-Medium	92.8 (+0.1)	68.9 (+0.3)	43.2 (+25.1)	7.4 (+2.5)
RF-DETR-Large	**95.6** (+0.1)	**86.3** (+0.2)	111.6 (+70.0)	17.2 (+7.4)

Switching to FP16 (Table 1) results in substantial inference speedup, while mAP@0.5 differences are negligible (typically <0.7 percentage points). This supports FP16 as the preferred choice for real-time deployment, combining faster inference and lower memory use without practical accuracy loss.

Appendix C. User experience survey examples

To provide full context for the user study, this appendix shows all survey examples. Each fragment was presented in three versions: (A) no overlay, (B) static labels, (C) 3D pose-based AR overlay (see Fig. C.16).

Appendix D. Fragment-level questionnaire results

See Fig. D.17.

Data availability

Data will be made available on request.

Fig. C.16. All survey examples used in the UX evaluation. Rows are fragments 1–4, columns are overlay versions A–C.

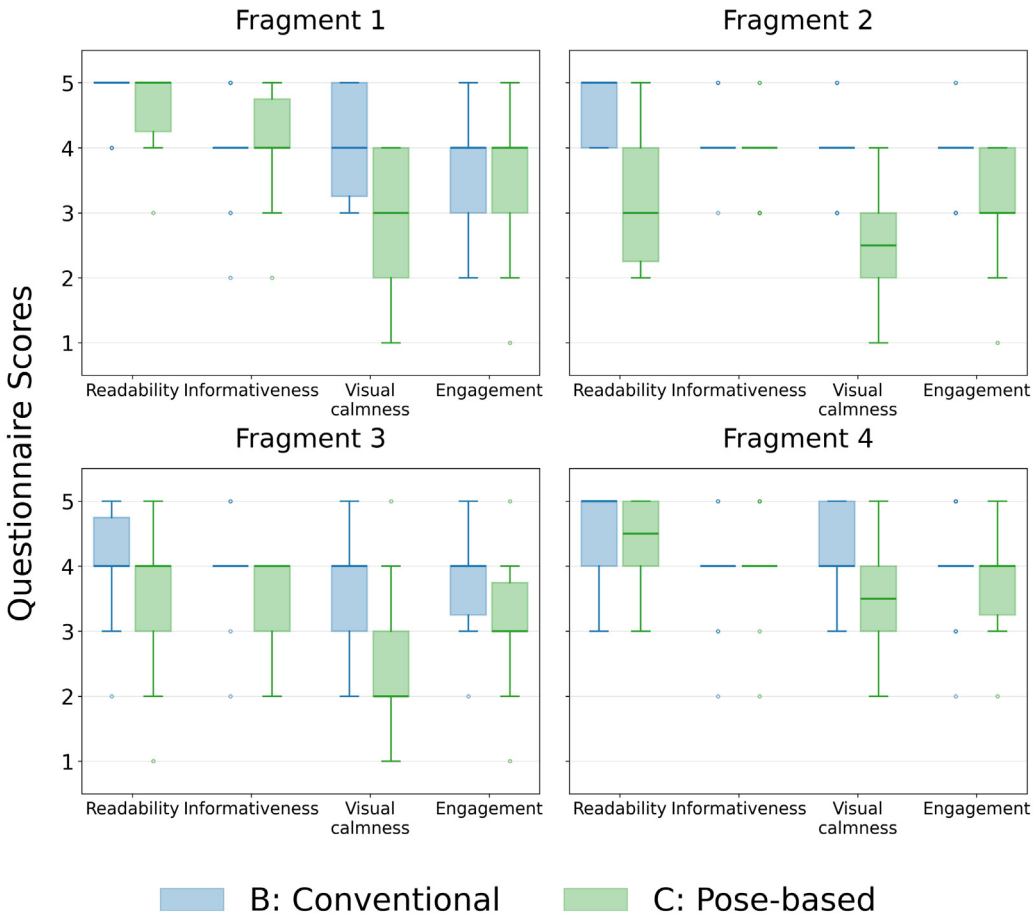


Fig. D.17. Boxplots of questionnaire scores per fragment, comparing overlay B (Conventional) and C (Pose-based) across the four UX dimensions (readability, informativeness, visual calmness, and engagement). Each subplot corresponds to one fragment.

References

Aharon, N., Orfaig, R., Bobrovsky, B.-Z., 2022. BoT-SORT: Robust associations multi-pedestrian tracking. arXiv:2206.14651. URL https://arxiv.org/abs/2206.14651.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP J. Image Video Process. 2008, http://dx.doi.org/10.1155/2008/246309.

Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016. Simple online and real-time tracking. CoRR abs/1602.00763. arXiv:1602.00763. URL http://arxiv.org/abs/1602.00763.

Bradski, G., 2000. The OpenCV library. Dr. Dobb's J. Softw. Tools.

Braun, M., Krebs, S., Flohr, F., Gavrila, D.M., 2018. The EuroCity persons dataset: A novel benchmark for object detection. CoRR abs/1805.07193. arXiv:1805.07193. URL http://arxiv.org/abs/1805.07193.

Broström, M., BoxMOT: pluggable SOTA tracking modules for object detection, segmentation and pose estimation models. doi:https://zenodo.org/record/7629840. URL https://github.com/mikel-brostrom/boxmot.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: Fast and flexible image augmentations. Information 11 (2), http://dx.doi.org/10.3390/info11020125, URL https://www.mdpi.com/2078-2489/11/2/125.

Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K., 2023. Observation-centric SORT: Rethinking SORT for robust multi-object tracking. arXiv:2203.14360. URL https://arxiv.org/abs/2203.14360.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. arXiv:2005.12872. URL https://arxiv.org/abs/2005.12872.

Clinckemaillie, W., Vanhaeverbeke, J., Slembrouck, M., Verstockt, S., 2026. One-shot team recognition and 3D pose estimation of cyclists for augmented reality visualization. In: Dong, J.-s., Sun, J., Xie, X., Jiang, K. (Eds.), Sports Analytics. Springer Nature Switzerland, Cham, pp. 36–52.

Corporation, N., 2025. NVIDIA TensorRT. URL https://developer.nvidia.com/tensorrt.

Cui, Y., Zeng, C., Zhao, X., Yang, Y., Wu, G., Wang, L., 2023. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. arXiv:2304.05170. URL https://arxiv.org/abs/2304.05170.

De Bock, J., Verstockt, S., 2021. Video-based analysis and reporting of riding behavior in cyclocross segments. Sensors 21 (22), 13, URL https://doi.org/10.3390/s21227619.

Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L., 2020. MOT20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003. URL https://arxiv.org/abs/2003.09003.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H., 2023. StrongSORT: Make DeepSORT great again. arXiv:2202.13514. URL https://arxiv.org/abs/2202.13514.

Duque Domingo, J., Medina Aparicio, R., González Rodrigo, L.M., 2021. Improvement of one-shot-learning by integrating a convolutional neural network and an image descriptor into a siamese neural network. Appl. Sci. 11 (17), http://dx.doi.org/10.3390/app11177839, URL https://www.mdpi.com/2076-3417/11/17/7839.

Dwivedi, S.K., Sun, Y., Patel, P., Feng, Y., Black, M.J., 2024. TokenHMR: Advancing human mesh recovery with a tokenized pose representation. arXiv:2404.16752. URL https://arxiv.org/abs/2404.16752.

Goebert, C., 2020. Augmented reality in sport marketing: Uses and directions. Sport. Innov. J. 1, 134–151. http://dx.doi.org/10.18060/24227.

Jocher, G., Qiu, J., 2024. Ultralytics YOLO11. URL https://github.com/ultralytics/ultralytics.

Jocher, G., Qiu, J., Chaurasia, A., 2023. Ultralytics YOLO. URL https://github.com/ultralytics/ultralytics.

Jonathon Luiten, A.H., 2020. TrackEval. https://github.com/JonathonLuiten/TrackEval.

Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J., 2018. End-to-end recovery of human shape and pose. arXiv:1712.06584. URL https://arxiv.org/abs/1712.06584.

Kocabas, M., Huang, C.-H.P., Hilliges, O., Black, M.J., 2021. PARE: Part attention regressor for 3D human body estimation. arXiv:2104.08527. URL https://arxiv.org/abs/2104.08527.

Koch, G.R., 2015. Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop. URL https://api.semanticscholar.org/CorpusID:13874643.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K., 2015. MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942. URL https://arxiv.org/abs/1504.01942.

Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., Gavrila, D.M., 2016. A new benchmark for vision-based cyclist detection. In: 2016 IEEE Intelligent Vehicles Symposium. IV, pp. 1028–1033. http://dx.doi.org/10.1109/IVS.2016.7535515.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P., 2015. Microsoft COCO: Common objects in context. arXiv:1405.0312.

Liu, H., Bhanu, B., 2019. Pose-guided R-CNN for jersey number recognition in sports. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 2457–2466. http://dx.doi.org/10.1109/CVPRW.2019.00301.

Ludwig, K., Lorenz, J., Kienzle, D., Bui, T., Lienhart, R., 2025a. Leveraging anthropometric measurements to improve human mesh estimation and ensure consistent body shapes. arXiv:2409.17671. URL https://arxiv.org/abs/2409.17671.

Ludwig, K., Oksymets, Y., Schön, R., Kienzle, D., Lienhart, R., 2025b. Efficient 2D to full 3D human pose uplifting including joint rotations. arXiv:2504.09953. URL https://arxiv.org/abs/2504.09953.

Luiten, J., 2021. How to evaluate tracking with the HOTA metrics. https://jonathonluiten.medium.com/how-to-evaluate-tracking-with-the-hota-metrics-754036d183e1, Medium blog post.

Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B., 2020. HOTA: A higher order metric for evaluating multi-object tracking. Int. J. Comput. Vis. 1–31.

Maggiolino, G., Ahmad, A., Cao, J., Kitani, K., 2023. Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification. arXiv:2302.11813. URL https://arxiv.org/abs/2302.11813.

Mashable, 2022. The yellow first-down line: an oral history of a game changer. URL https://mashable.com/archive/yellow-first-down-line.

Naik, B.T., Hashmi, M.F., Bokde, N.D., 2022. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. Appl. Sci. 12 (9), http://dx.doi.org/10.3390/app12094429, URL https://www.mdpi.com/2076-3417/12/9/4429.

Noble, J., 2023. Ai replays and augmented reality: What's new for F1's TV coverage in 2023. URL https://www.autosport.com/f1/news/ai-replays-and-more-augmented-reality-whats-new-for-f1s-tv-coverage-in-2023/10439342/.

Owens, N., Harris, C., Stennett, C., 2003. Hawk-Eye tennis system. In: International Conference on Visual Information Engineering. pp. 182–185. http://dx.doi.org/10.1049/cp:20030517.

Pei-Xia, S., Hui-Ting, L., Luo, T., 2016. Learning discriminative CNN features and similarity metrics for image retrieval. pp. 1–5. http://dx.doi.org/10.1109/ICSPCC.2016.7753634.

Ren, H., 2022. Sports video athlete detection based on deep learning. Neural Comput. Appl. 35, 4201–4210, URL https://api.semanticscholar.org/CorpusID:247918422.

Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. arXiv:1609.01775. URL https://arxiv.org/abs/1609.01775.

Robicheaux, P., Popov, M., Madan, A., Robinson, I., Nelson, J., Ramanan, D., Peri, N., 2025. Roboflow100-VL: A multi-domain object detection benchmark for vision-language models. arXiv:2505.20612. URL https://arxiv.org/abs/2505.20612.

Robinson, I., Robicheaux, P., Popov, M., Ramanan, D., Peri, N., 2025. RF-DETR. https://github.com/roboflow/rf-detr, SOTA Real-Time Object Detection Model.

Santos, P.D., Jerri, V., 2023. Team Recognition in Sports Events (Master's thesis). Polytechnic University of Catalonia.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. FaceNet: A unified embedding for face recognition and clustering. CoRR abs/1503.03832. arXiv:1503.03832. URL http://arxiv.org/abs/1503.03832.

Scott, A., Uchida, I., Ding, N., Umemoto, R., Bunker, R., Kobayashi, R., Koyama, T., Onishi, M., Kameda, Y., Fujii, K., 2024. TeamTrack: A dataset for multi-sport multi-object tracking in full-pitch videos. arXiv:2404.13868. URL https://arxiv.org/abs/2404.13868.

Sekachev, B., Manovich, N., Zhiltsov, M., Zhavoronkov, A., Kalinin, D., Hoff, B., TOsmanov, Kruchinin, D., Zankevich, A., DmitriySidnev, Markelov, M., Johannes222, Chenuet, M., a andre, telenachos, Melnikov, A., Kim, J., Ilouz, L., Glazov, N., Priya4607, Tehrani, R., Jeong, S., Skubriev, V., Yonekura, S., truong, v., zliang7, lizhming, Truong, T., 2020. Opencv/cvat: v1.1.0. http://dx.doi.org/10.5281/zenodo.4009388.

Stanojevic, V.D., Todorovic, B.T., 2024. BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking. Mach. Vis. Appl. 35 (3), http://dx.doi.org/10.1007/s00138-024-01531-5.

Supponor, 2020. Cutting Through The Regulation of The Virtual Advertising Landscape. Tech. rep., Supponor Ltd., (Accessed 14 January 2025). URL https://supponor.com/whitepaper-cutting-through-the-regulation-of-the-virtual-advertising-landscape/.

Verstockt, S., Van den broeck, A., Van Vooren, B., De Smul, S., De Bock, J., 2020. Data-driven summarization of broadcasted cycling races by automatic team and rider recognition. In: Proceedings of the 8th International Conference on Sport Sciences Research and Technology Support - icSPORTS. INSTICC, SciTePress, pp. 13–21. http://dx.doi.org/10.5220/0010016900130021.

Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. CoRR abs/1703.07402. arXiv:1703.07402. URL http://arxiv.org/abs/1703.07402.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022. ByteTrack: Multi-object tracking by associating every detection box. arXiv:2110.06864. URL https://arxiv.org/abs/2110.06864.

Zhang, R., Wu, L., Yang, Y., Wu, W., Chen, Y., Xu, M., 2020. Multi-camera multi-player tracking with deep player identification in sports video. Pattern Recognit. 102, 107260. http://dx.doi.org/10.1016/j.patcog.2020.107260, URL https://www.sciencedirect.com/science/article/pii/S0031320320300650.

Zhou, K., Yang, Y., Cavallaro, A., Xiang, T., 2019. Omni-scale feature learning for person re-identification. arXiv:1905.00953. URL https://arxiv.org/abs/1905.00953.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2021. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159. URL https://arxiv.org/abs/2010.04159.