DSAI 544 Project Report
Basketball Player Tracking

Ahmet Yusuf Öztürk

January 15, 2026

Abstract

The objective of this project is to design a computer vision pipeline for automated sports narration. We specifically designed it for use in EuroLeague Basketball games. Our motivation was to help visually impaired fans, inspired from Görmezden Gelme project by 1907 UNIFEB. This project aims to detect and track players in dynamic, crowded environments distinct from NBA settings. The methodology contains three major components, which are defined in the project scope: Dataset creation, Auto-Labeling, and Model Fine-Tuning. We have generated a dataset by extracting frames from EuroLeague match footage *. These frames were later annotated using an auto-labelling pipeline; we first used Grounding DINO to auto-label the images, and after not being satisfied with the results, we decided to move on with SAM3. We then fine-tuned a YoloV8m model on this annotated dataset. We finally tested the model using unseen clips from the same game.

1 Introduction

As computer vision technologies and AI advance, they are involved in every area of our lives. The Sports Industry is one of those areas and has huge potential to benefit from those technologies. While there are already significant works have been done, e.g. Basketball AI, Euroleague games present unique challanges. The Euroleague courts are smallert than the NBA court, which results in higher player density, and more frequent occlusions which results in distinct visual patterns.

The primary motivation for this project is social accessibility. As a former member of 1907 UNIFEB Boğaziçi (a student fan club for Fenerbahçe), I participated in narrating games for visually disabled fans. This manual process limits scalability. The ultimate goal of this research is to automate this narration pipeline, allowing more visually impaired fans to enjoy matches in real-time in stadiums. By developing a robust vision model capable of ”seeing” the game, we can eventually feed these events into a Generative AI to produce natural language commentary.

2 Methods and Results

2.1 Dataset Creation

To address the specific visual conditions of the EuroLeague, I implemented the **Dataset Creation** component.

- **Source Material:** A total of 5 minutes and 36 seconds 12 gameplay footage from a Euroleague game was processed.
- **Frame Extraction:** Frames were extracted at 0.5-second intervals to ensure diversity while maintaining continuity. This resulted in a dataset of 672 images.

2.2 Auto-Labelling

For the **Auto-Labelling** component, we tried two teacher models and pick the best performing one:

- **Grounding DINO:** We first tried using GroundiNG DINO as our teacher model, however the model frequently misclassified objects and struggled with the visual clutter of the smaller court. Visual inspection revealed 3-4 major errors per frame (e.g., confusing referees with players or falsely address players' hands as a ball), making manual correction of 600+ frames infeasible.
- **SAM3:** Consequently, we switched the pipeline to SAM3. This model provided significantly higher quality annotations, which were then used to create the training set for the fine-tuning part.

After determining the teacher model, we detect object belonging to **7** main classes:

- Player
- Jersey Number
- Referee
- Ball
- Basketball Rim
- Ad Basketball
- Shot Clock

The model seems to confuse the real ball played with the balls in the adversary boards by the pitch, so we have decided to add an *Ad Basketball* class to prevent this confusion. With similar reasoning we added the class *Shot Clock*. Remaining classes were added to determine the players, referee, the jersey number (player id), and keep tracking of the events.

2.3 Model Fine-Tuning

For the fine-tuning part, we have selected the YoloV8m (You look only once, medium) architecture. This architecture provides low latency and high detection accuracy. We then fine-tuned the model using a custom annotated dataset, specifically training it to recognise key classes such as players, referees, and the ball within the specific lighting and court design of EuroLeague stadiums.

2.4 Experimental Results

The model's performance was evaluated using standard metrics we tracked during the training process.

- Training Metrics: The loss curves and mean Average Precision (mAP) scores indicate the model successfully converged.

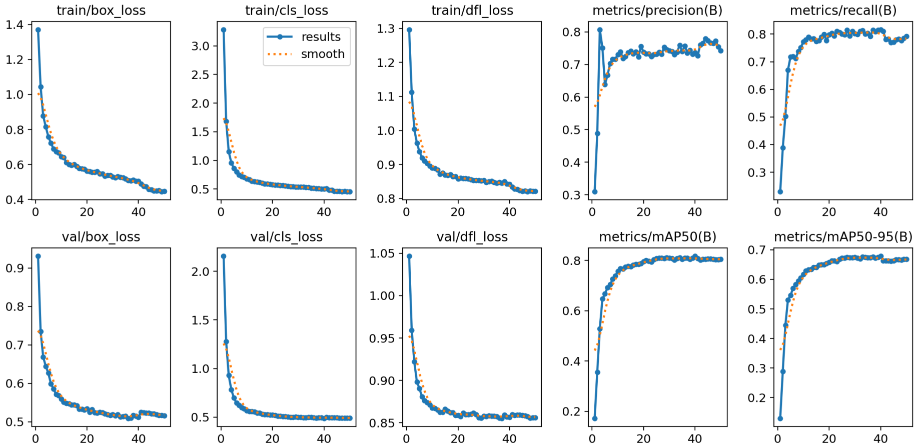

Figure 1: Training and Validation loss curves showing model convergence

- **Qualitative Testing:** The fine-tuned model was then tested on a sample video clip from the same game, which was excluded from training and validation sets. The model successfully detected the classes; more importantly, it showed us its capability of doing such detections in real time. One can see the model's performance on the video.

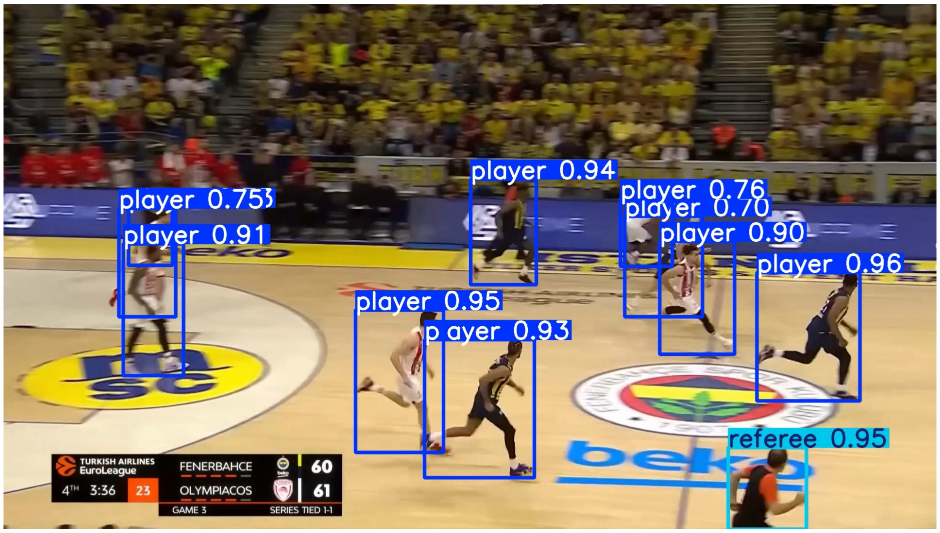

Figure 2: Sample frame from the resulting clip.

3 Conclusion

We successfully showed the application of computer vision techniques to solve a real-world problem: tracking players in Euroleague basketball. By creating a Dataset, utilizing Auto-Labeling with SAM3 to overcome the limitations of Grounding DINO, and fine-tuning a YOLOv8m model, we established a robust detection baseline.

3.1 Future Work and Improvements

While the detection pipeline is functional, there still exist several limitations. We will work on those to make this project better; however, due to time limitations, this project is submitted in its current state.

Future works:

- Jersey Number Recognition: We have attempted to tracked player's from their jersey numbers, the model has not succeeded in correctly detecting jersey numbers from the training images, and we changed our direction to only detect players. Future work would involve a two-stage pipeline: Cropping the detected player first, then running OCR or a classification model on the crop.
- The model can detect players and assign players the same id as long as the player can be shown on the screen. However, if a player disguises and reappers the model assigns that player a different id *see. After implementing the jersey number detection, we will classify players into 100 classes ranging from 0 to 99, and the model will learn the players.
- We also need to cluster players in two teams so that two players with the same jersey number will be differentiated.
- To fully replicate the "Basketball AI" inspiration, we need to implement homography to map 2D pixel coordinates to a top-down view of the court.
- By completing those steps, we can finally implement a model which will log game events, so that we can feed our narrator with those logs.

4 References

- Finetuning YoloV8 model Youtube
- Basketball AI
- Grounding DINO Usage
- SAM3 Usage