Final Report: Player Re-Identification in Sports Footage

By Bandari Harshadeep Reddy

1. Objective

The objective of this assignment was to identify and re-identify players in a 15-second soccer video using object detection and tracking techniques. The task involves assigning consistent IDs to players even when they go out of frame and return, simulating a real-time re-identification system.

2. Approach & Methodology

a. Object Detection

- I used the provided “best.pt” model, a fine-tuned version of **Ultralytics YOLOv11**.
- Frames were resized from 1280x720 to 480x270 for faster processing.
- Detection was run on every frame using the ultralytics Python API.

b. Tracking and Re-identification

- I integrated **DeepSORT**, a widely-used multiple objects tracking algorithm.
- The original **ResNet embedder** was replaced with **MobileNet** to reduce inference time.
- DeepSORT tracks were maintained using appearance embeddings and Kalman filtering.

c. Consistency Handling

- IDs were preserved even if a player left the frame and reappeared later.
- Visualization included consistent bounding boxes, ID labels, FPS and frame counters.

3. Codebase Structure

player-reid-single-feed

➢ input/	contains video and best.pt model
➢ output/	generated output video
➢ src/	main scripts (reid_main.py, detector.py, tracker.py)
➢ helpers/	draw utilities (draw_boxes.py) and an empty(__init__.py) file.
➢ README.md	documentation and usage guide

4. Techniques Tried and Outcomes

Technique	Outcome
Frame Resizing (480x270)	2–3x faster processing time
MobileNet Embedder	Faster runtime than ResNet (with minimal loss)
Consistent Colour IDs	Easier tracking of individual players visually
Label Overlays	Improved readability during fast player motion
FPS + Frame Display	Helped evaluate processing performance

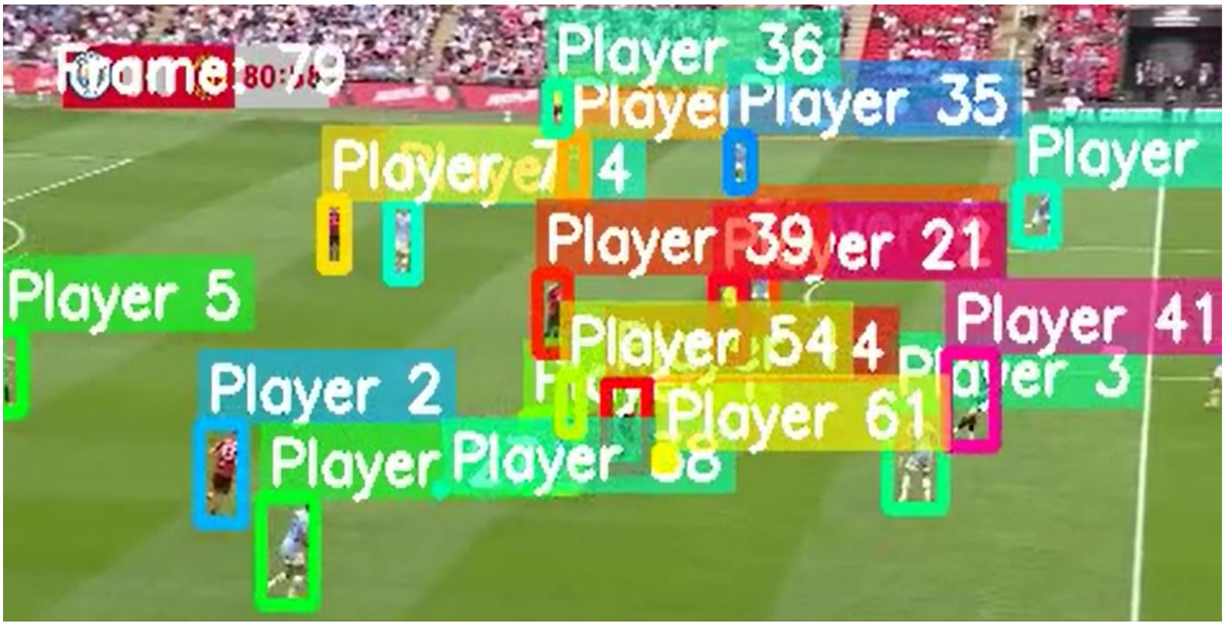
5. Challenges and Solutions

Challenge	Solution
Runtime too high on CPU	Resized input + used MobileNet embedder
Colab OpenCV windows not supported	Used IPython display with HTML5 video rendering
Duplicate IDs during fast re-entries	Tuned DeepSORT parameters and used visual inspection

6. Results

- **Total Frames Processed**: 375
- **Runtime (Colab CPU)**: ~47 seconds
- **Effective FPS**: ~7.9
- **Model Used**: Fine-tuned YOLOv11 (best.pt)
- **Tracker**: DeepSORT with MobileNet Embedder
- **Output Video**: output/fixed_output.mp4

7. Sample Frame



8. Links

- **GitHub Repository**: https://github.com/Harshadeep3063/player-reid-single-feed
- **Model Download (best.pt)**: best.pt

9. Conclusion

This assignment gave me hands-on experience in object detection, tracking, and re-identification — combining theory with practical model integration. I improved runtime significantly through embedding and resolution optimization and learned to manage ML repositories for real-world constraints like GitHub size limits.

I am confident this approach can scale to longer videos, multi-camera feeds, or vision-language integrations with further work.

10. Author

Bandari Harshadeep Reddy
A recent B.Tech Graduate– Computer Science (Data Science)
CVR College of Engineering
✉ harshadeepreddybandari@gmail.com
🔗 github.com/Harshadeep3063