

Task-1: Cross Camera Player Mapping

1. Approach and Methodology

- **Object Detection:** Used a fine-tuned YOLOv11 model ([best.pt](#)) to detect players in both the broadcast and tacticam video feeds.
- **Feature Extraction:** Leveraged pre-trained **ResNet50** (excluding final FC layer) to extract robust visual embeddings for each detected player.
- **Matching Strategy:**
 - Calculated **cosine similarity** between embeddings of players across views.
 - Added **spatial information** (normalized distance between player centers) for more robust mapping.
 - Combined **visual + spatial similarity** using a weighted sum for final similarity matrix.
- **Result Output:** Saved matches in a structured **JSON format** with frame numbers, bounding boxes, player IDs, and similarity scores.

2. Techniques Tried

- YOLOv11 for object detection
- ResNet50 for visual feature representation
- Cosine similarity for embedding comparison
- Center-based spatial similarity for improving mapping robustness
- Vectorized matrix operations for speed
- JSON output format for reproducibility and automation

3. Challenges

- Matching across camera views where player scale/angle differs.
- Some bounding boxes returned blank crops → filtered out.
- Processing time was long on CPU optimized by printing frame status and switching to GPU.
- Cross-view variation (scale, pose, lighting) made matching non-trivial.
- Zero-area crops occurred due to invalid bounding boxes – filtered these before feature extraction.

4. Remaining Work (if any)

- Could add tracking (Deep SORT) to ensure frame-to-frame consistency.
- Could visualize matched players with bounding boxes overlayed in output videos.
- Could develop a better output representation where output could be viewed in GUI etc.