# Project Title: Cross-Camera Player Mapping

---

### Objective

The goal of this project is to assign consistent player IDs across two different camera views of the same gameplay:

- **`broadcast.mp4`**: Traditional game view from a high and wide angle.
- **`tacticam.mp4`**: Tactical low-angle or close-up camera capturing the same action.

Players must be matched across both feeds, even though:

- Camera angles differ
- Player positions may shift
- Appearances may slightly change

---

### Problem Statement

How do we ensure that:

Player A in broadcast.mp4 is matched correctly to the same Player A in tacticam.mp4, even though they appear different due to camera angles?

This is a cross-view re-identification (Re-ID) challenge, and we solve it using **visual embeddings**.

---

## Methodology & Approach

The solution is broken into the following stages:

### 1. Player Detection using YOLOv8

- We use a **custom-trained YOLOv8 model (`best.pt`)** to detect players in both videos.
- YOLO outputs bounding boxes and confidence scores for each detected player.
- Detections with confidence above 0.8 are considered valid.

### 2. Visual Embedding Extraction with ResNet50

- For each detected player (bounding box), we:
  - Crop the player from the frame.
  - Pass it through a **pretrained ResNet50** model (without the final classification layer).

### 3. Embedding Comparison Using Cosine Similarity

- Once we extract embeddings for all players from both videos:
  - We use **cosine similarity** to compare the visual features of each player in **tacticam.mp4** against every player in **broadcast.mp4**.
  - The player in `tacticam` is matched to the one in `broadcast` with the highest similarity score (above a defined threshold).

### 4. Player ID Mapping

- After similarity matching, we build a mapping:

```
mapping = {}

    for i, t_id in enumerate(t_ids):

                best_match_index = np.argmax(sim_matrix[i])

                   b_id = b_ids[best_match_index]

                        mapping[t_id] = b_id
```

- This ensures **consistent IDs** across camera angles.

---

## Techniques Tried and Outcomes

| Component | Technique Used | Result |
|---|---|---|
| Model | **YOLOv8 with `best.pt`** | Accurate bounding boxes |
| Feature Extraction | **ResNet50** | Good visual differentiation |
| Matching | **Cosine similarity** | Effective in clean visual cases |

---

## Challenges Encountered

| Issue | Description |
|---|---|
| Different camera angles | Same player can look different due to angle, lighting, and size variations |
| Player occlusion or blur | Visual features may degrade for partially visible players |
| Similar jerseys | Matching players from the same team can lead to incorrect mapping |
| Frame mismatch | Slight timing differences in frames between the two videos |

# Results

- **High accuracy** in matching players when:
  - Players are fully visible
  - Background contrast is decent
  - Players have distinct shapes or jersey numbers
- **Lower accuracy** when:
  - Players are occluded or heavily blurred
  - Frames are too different in timing