Report: Player Re-ID and Cross-Camera Mapping

Approach

Cross Camera Mapping

- **Used the YOLOv11 model to detect players in each frame from both camera angles (broadcast.mp4 and tacticam.mp4).**
- **Extracted appearance-based features (RGB color histograms) from each player's bounding box.**
- **Compared features using cosine similarity to match player identities across the two camera feeds.**
- **Created a consistent player ID mapping for each individual from broadcast to tacticam view.**

```
PS C:\Users\bhava\Downloads\Assignment> python cross_camera.py
Creating new Ultralytics Settings v0.0.6 file
View Ultralytics Settings with 'yolo settings' or at 'C:\Users\bhava\AppData\Roaming\Ultralytics\settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultra
lytics-settings.

0: 384x640 3 players, 1308.6ms
Speed: 9.4ms preprocess, 1308.6ms inference, 40.6ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 players, 898.2ms
Speed: 3.1ms preprocess, 898.2ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 4 players, 901.8ms
Speed: 2.0ms preprocess, 901.8ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 players, 891.0ms
Speed: 2.3ms preprocess, 891.0ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 4 players, 864.9ms
Speed: 2.1ms preprocess, 864.9ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 11 players, 1 referee, 871.2ms
Speed: 2.6ms preprocess, 871.2ms inference, 6.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 goalkeeper, 12 players, 1 referee, 892.3ms
Speed: 2.6ms preprocess, 892.3ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 1 goalkeeper, 14 players, 1 referee, 905.9ms
Speed: 2.6ms preprocess, 905.9ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 1 goalkeeper, 13 players, 1 referee, 904.6ms
Speed: 2.6ms preprocess, 904.6ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 1 goalkeeper, 13 players, 1 referee, 892.5ms
Speed: 2.5ms preprocess, 892.5ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 1 goalkeeper, 12 players, 1 referee, 911.1ms
Speed: 2.8ms preprocess, 911.1ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 1 goalkeeper, 13 players, 1 referee, 868.5ms
Speed: 2.7ms preprocess, 868.5ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)
```

Single Feed Re-Identification

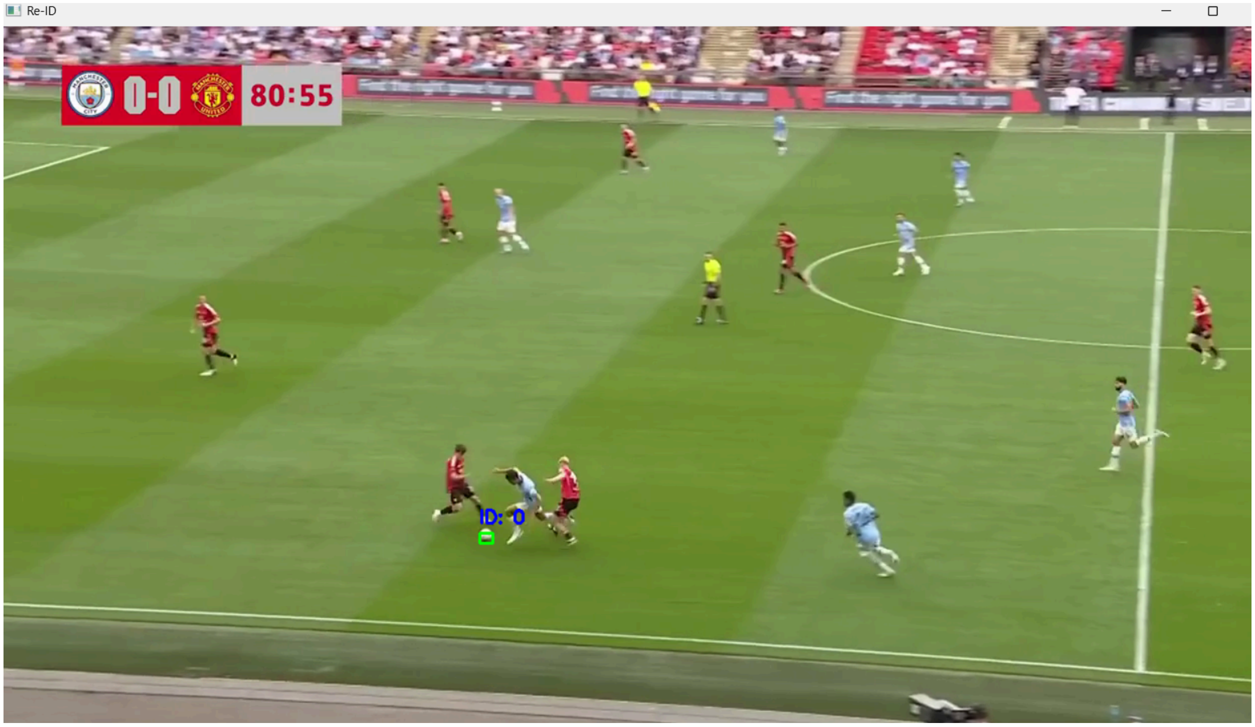
- **Used YOLOv11 to detect players in all frames of a single video feed (15sec_input_720p.mp4).**

- **Maintained identity consistency over time by matching re-appearing players based on their visual appearance.**

- **Implemented a feature-based matching mechanism to reassign previous IDs using cosine similarity on color histograms.**

- **Ensured player IDs remained consistent even when players exited and re-entered the frame.**

```
PS C:\Users\bhava\Downloads\Assignment> python reid_single_feed.py

0: 384x640 1 ball, 16 players, 2 referees, 756.4ms
Speed: 2.2ms preprocess, 756.4ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 18 players, 2 referees, 957.1ms
Speed: 3.7ms preprocess, 957.1ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 16 players, 2 referees, 907.4ms
Speed: 1.9ms preprocess, 907.4ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 14 players, 2 referees, 863.1ms
Speed: 2.1ms preprocess, 863.1ms inference, 1.4ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 14 players, 2 referees, 903.4ms
Speed: 2.2ms preprocess, 903.4ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 16 players, 2 referees, 923.4ms
Speed: 1.9ms preprocess, 923.4ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 15 players, 2 referees, 898.9ms
Speed: 2.1ms preprocess, 898.9ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 15 players, 1 referee, 920.9ms
Speed: 2.7ms preprocess, 920.9ms inference, 1.6ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 16 players, 1 referee, 913.9ms
Speed: 1.9ms preprocess, 913.9ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 15 players, 2 referees, 891.2ms
Speed: 2.7ms preprocess, 891.2ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 15 players, 2 referees, 937.2ms
Speed: 2.2ms preprocess, 937.2ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 15 players, 2 referees, 893.0ms
Speed: 1.9ms preprocess, 893.0ms inference, 1.4ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 ball, 13 players, 2 referees, 855.7ms
Speed: 2.1ms preprocess, 855.7ms inference, 1.4ms postprocess per image at shape (1, 3, 384, 640)
```



Techniques Explored

- **Feature Extraction: RGB color histogram features from player bounding boxes.**

- **Matching Strategy: Cosine similarity used to associate player features across frames or feeds.**

- **Baseline Tracking: No use of Kalman filter or motion models; relied solely on visual similarity.**

Challenges

- **Occlusion and overlapping players sometimes led to incorrect ID assignment.**

- **Uniform similarity (e.g., identical jerseys) made visual features less effective for matching.**

- **No motion modeling made it hard to track players during fast-paced transitions or crowding.**

Future Improvements

- **Integrate a tracking system like Deep SORT or a transformer-based tracker for motion-aware tracking.**
- **Replace histograms with learned embeddings from a pretrained re-ID model like OSNet or ResNet-50.**
- **Add temporal smoothing, trajectory prediction, and spatial constraints to boost consistency.**
- **Evaluate results using standard re-ID metrics like mAP, IDF1, and TrackEval.**