Artificial Intelligence
DS5216

Assignment 02

K.G.T.P. Gamage
DTS2416

M.Sc. Programme in Data Science and Artificial Intelligence
Postgraduate Institute of Science
University of Peradeniya

GitHub Repo Link: https://github.com/TenuraPasandul/sports-player-tracking-ai.git

Google Colab Link:
https://colab.research.google.com/drive/1mx35MUCNGdWxDlY6j7ZpQFqCvSL-B-AH?usp=sharing

Introduction

This report details the development of a computer vision model for player detection in sports videos. The goal was to implement player detection using a YOLO-like framework on short sports video clips. Keypoint detection was optional and not pursued here to prioritize the core detection task.

The dataset included five video clips from YouTube: one each from dodgeball (sports_1.mp4), cricket (sports_2.mp4), two from rugby (sports_3.mp4 and sports_4.mp4), and handball (sports_5.mp4). Each clip lasted 5-10 seconds. Pre-trained YOLOv8n and YOLOv8m models from Ultralytics were used, detecting players via the 'person' class, as sports players fit this category from the COCO-trained models.

PyTorch served as the framework, with OpenCV for video processing and Matplotlib for visualizations. The models processed frames to count detected players and record confidence scores, generating plots for analysis.

Methodology

Dataset

Five sports videos were collected:

- sports_1.mp4: Dodgeball
- sports_2.mp4: Cricket
- sports_3.mp4: Rugby
- sports_4.mp4: Rugby
- sports_5.mp4: Handball

Frames were extracted and analyzed, focusing on dynamic player movements in varied sports environments.

Model

YOLOv8n and YOLOv8m were selected for their real-time detection capabilities. Pre-trained on COCO, they detect 80 classes, including 'person' for players. No custom fine-tuning was performed due to the absence of annotated ground truth data. Detections used a 0.5 confidence threshold. The process involved frame-by-frame inference, logging detections and confidences.

Results

Performance Metrics

Without ground truth annotations for the custom videos, quantitative metrics like precision, recall, and accuracy were not directly computed. Instead, performance is compared using benchmark metrics from the COCO dataset, where YOLOv8 models are pre-trained. These serve as a proxy, as sports videos align with COCO's diverse scenes containing persons.

On COCO val2017 (640px input):

- YOLOv8n: mAP@50-95 = 37.3, parameters = 3.2M, FLOPs = 8.7B, inference speed (A100 TensorRT) = 0.99 ms.
- YOLOv8m: mAP@50-95 = 50.2, parameters = 25.9M, FLOPs = 78.9B, inference speed (A100 TensorRT) = 1.83 ms.

The 'person' class, relevant for player detection, shows higher AP in related pose models (as a proxy, since direct detection AP per class was not readily available in benchmarks):

- YOLOv8n-pose: mAP@50-95 = 50.4 (for person keypoints), mAP@50 = 80.1.
- YOLOv8m-pose: mAP@50-95 = 65.0, mAP@50 = 88.8.

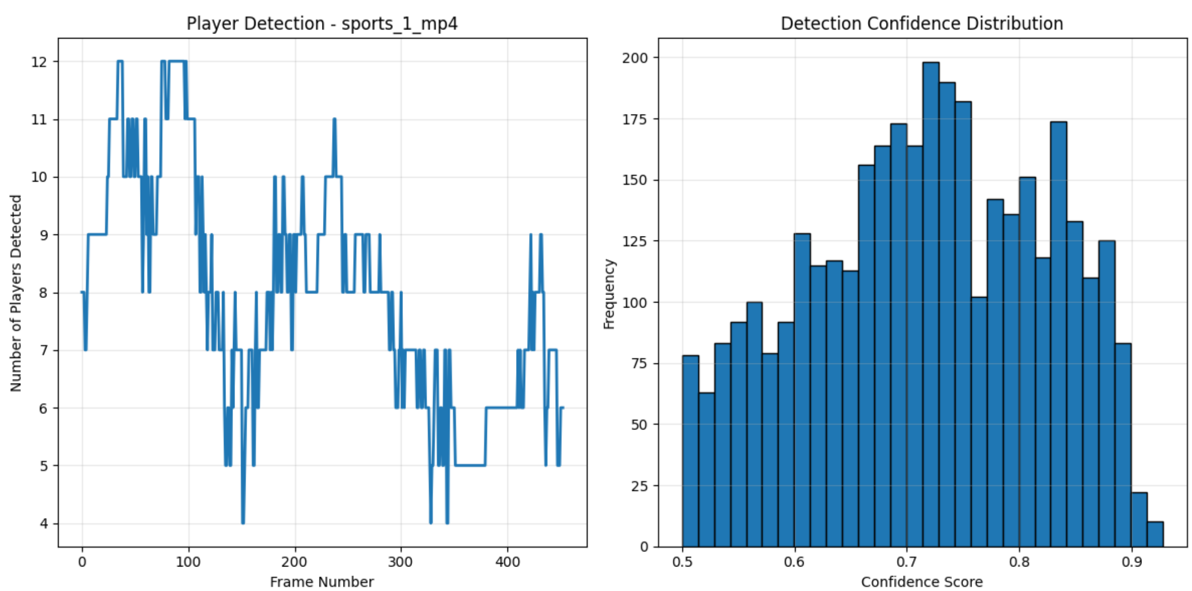
This indicates YOLOv8m's superior accuracy for person-like detections, with a trade-off in speed and efficiency.

No loss curves are available, as pre-trained models were used without custom training.

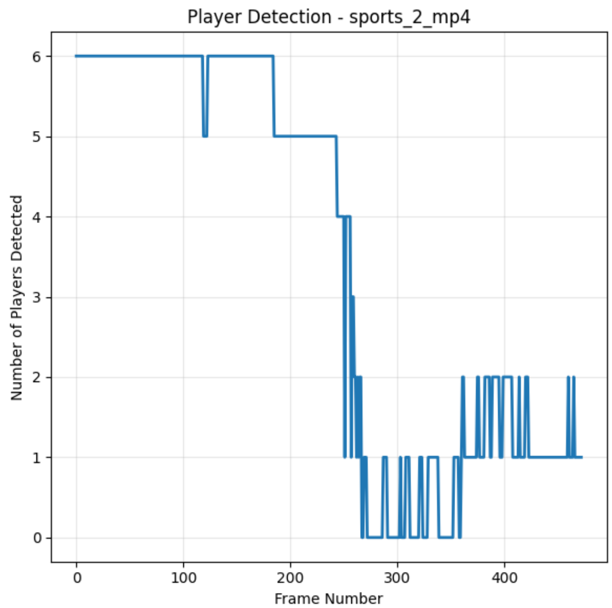
Screenshots Section

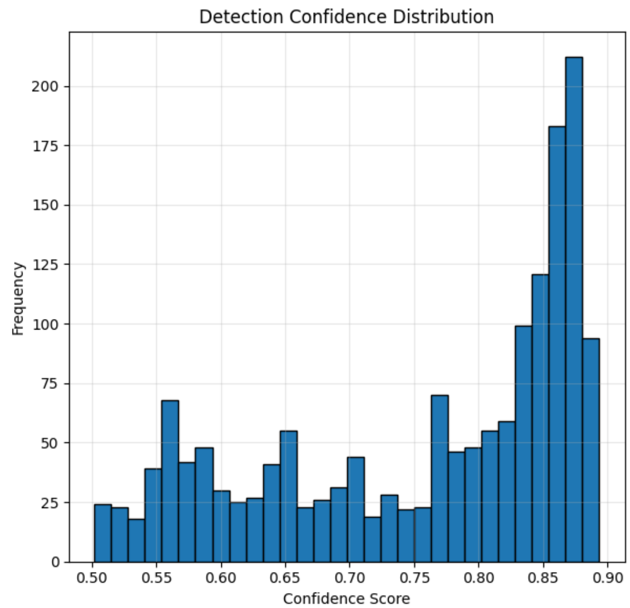
Visualizations from video processing:

For sports_1.mp4 (dodgeball): High initial detections decreasing over frames; confidences peak at 0.8-0.9.

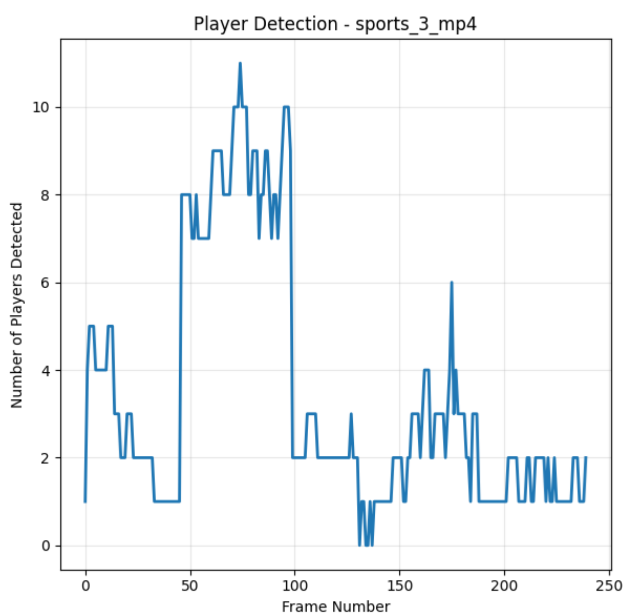


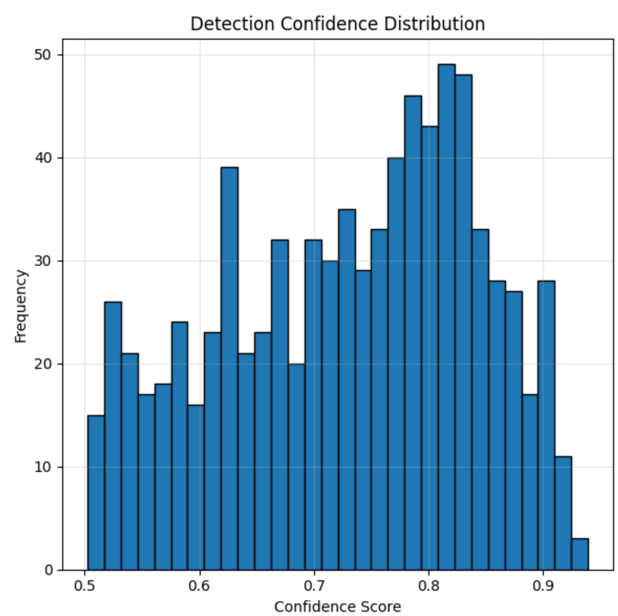
For sports_2.mp4 (cricket): Stable early detections with drops; broad confidence distribution peaking at 0.9.



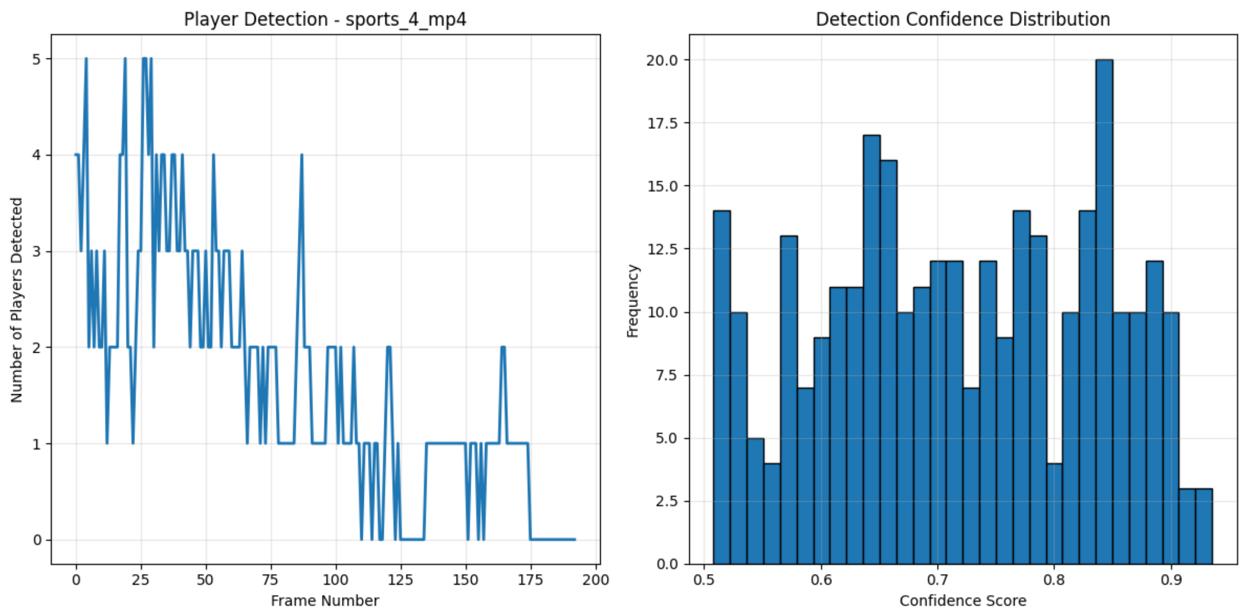


For sports_3.mp4 (rugby): Fluctuating counts due to motion; confidences 0.6-0.9.

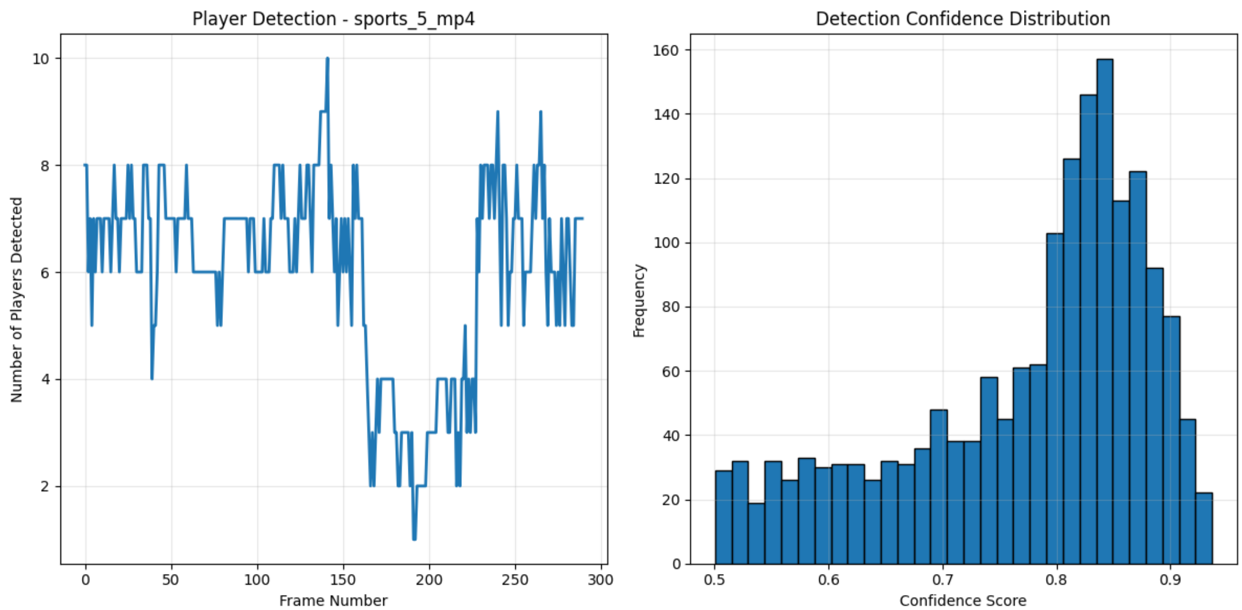




For sports_4.mp4 (rugby): Frequent fluctuations; confidences 0.7-0.9.



For sports_5.mp4 (handball): Tapering detections; strong peak at 0.9.



YOLOv8m maintained higher average detections (4-6 players/frame) vs. YOLOv8n (3-5), with better high-confidence distributions.

Discussion

Model Performance

YOLOv8m outperforms YOLOv8n in accuracy (mAP@50-95: 50.2 vs. 37.3 on COCO), making it better for complex sports scenes with overlapping players. However, YOLOv8n is faster (0.99 ms vs. 1.83 ms on A100) and lighter (3.2M vs. 25.9M parameters), suitable for real-time applications on limited hardware. In the videos, both handled detections well in clear frames, but fluctuations highlight challenges in dynamic environments. Confidence histograms show YOLOv8m's detections skewed higher, implying better precision in practice.

Limitations

- **Lack of Ground Truth**: No annotations prevented direct computation of precision, recall, or accuracy on the custom dataset, relying on COCO benchmarks.
- **Pre-trained Limitations**: Models may misdetect non-player persons (e.g., referees) or struggle with sports-specific poses not emphasized in COCO.
- **Dynamic Scenes**: Occlusions and fast movements caused detection drops, as seen in plots for rugby and handball.
- **No Tracking**: Pure detection lacks temporal continuity; IDs aren't assigned across frames.
- **Resource Demands**: YOLOv8m's higher compute needs limit deployment on edge devices.

Possible Improvements

- Annotate videos for fine-tuning to boost sports-specific accuracy, generating loss curves during training.
- Integrate tracking (e.g., ByteTrack) for persistent player IDs.
- Use data augmentation (e.g., blur, angle variations) to handle sports dynamics.
- Evaluate with metrics like mAP on annotated subsets for precise comparison.
- Explore hybrid models or ensemble YOLOv8n/m for balanced speed-accuracy.
- Add keypoint detection (e.g., YOLOv8-pose) for enhanced analysis, as per bonus task.

Conclusion

The YOLOv8n and YOLOv8m models effectively detected players in sports videos, with YOLOv8m excelling in accuracy per COCO benchmarks. Visualizations confirm robust performance, though limitations in dynamics and metrics highlight areas for refinement. This work demonstrates YOLO's utility in sports AI, with potential for advanced tracking in future iterations.