Capstone Project Report: Real-Time Fatigue Monitoring in Basketball Using Computer Vision

# 1. Introduction

Fatigue plays a pivotal role in influencing athletic performance, injury risk, and decision-making ability in high-speed sports such as basketball. Despite the availability of wearable-based fatigue tracking technologies, their high cost and invasiveness can make them impractical in many team or amateur settings. This project introduces a vision-based, real-time player fatigue detection system using only game video footage. By leveraging object detection and tracking, the system aims to estimate fatigue indicators dynamically and non-invasively, offering an accessible tool for coaches and analysts.

# 2. Literature Review

Numerous studies have highlighted the significance of fatigue analysis in sports. According to Gabbett (2016), prolonged periods of high-speed activity are linked to increased injury risk, especially when rest and recovery are inadequate. Traditional methods, such as heart rate variability (HRV) and lactate threshold monitoring, though accurate, often require wearable sensors and cannot be integrated with video-based match review systems. Recent work by Pons et al. (2020) suggests the potential of motion-based fatigue proxies like deceleration events, speed variability, and time-on-field in video analysis. Our system builds on these insights by modeling fatigue through observable motion characteristics extracted from tracked video.

# 3. Project Overview

This project aims to develop a computer vision pipeline that detects players, tracks them across frames, and estimates their fatigue using motion-based metrics. The final result is an annotated video that overlays bounding boxes, fatigue levels, and real-time statistics for each player.

# 4. Tools and Technologies

- Programming Language: Python  
- Object Detection: YOLOv8 (Ultralytics)  
- Tracking: Deep SORT  
- Dataset Annotation: Roboflow (YOLOv8 format)  
- Libraries: OpenCV, NumPy, Roboflow API  
- Platform: Google Colab / Jupyter Notebook

# 5. Dataset Preparation

The training dataset consisted of annotated basketball player frames prepared using Roboflow. The annotations included bounding boxes for players, exported in YOLOv8 format. Roboflow's workspace and API integration simplified the training pipeline and ensured dataset consistency. The annotated data enabled the YOLOv8 model to detect individual players in varied court settings.

# 6. Fatigue Modeling Methodology

Fatigue is computed from three real-time, motion-based indicators:  
- Speed Drop: Reflects decline in player velocity across recent frames  
- Speed Variability: Standard deviation of player speeds, indicating inconsistency  
- Time on Court: Total number of frames a player is tracked, converted to seconds  
These metrics are normalized and used to calculate a composite fatigue score. Based on this score, players are categorized into Low, Medium, or High fatigue levels. The system overlays this information visually using bounding box colors and fatigue bars.

# 7. System Architecture and Workflow

1. Input video is read frame-by-frame  
2. YOLOv8 detects players in each frame  
3. Deep SORT assigns consistent IDs to players across frames  
4. Motion features (position, speed) are tracked and stored  
5. Fatigue score is computed dynamically using recent frame data  
6. Video is overlaid with bounding boxes, fatigue bars, and a dashboard  
7. Annotated output is saved as a new video file

# 8. Visual Output Description

Each frame in the final video shows:  
- Player ID with bounding box  
- Fatigue level: Low (green), Medium (yellow), High (red)  
- Fatigue bar: Horizontal indicator representing score percentage  
- Real-time top 3 fatigued players  
- Bottom overlay showing frame number, player count, and average fatigue  
A fatigue spike warning appears when any player's score exceeds 0.6.

# 9. Evaluation

The model was tested on full-length basketball video footage with 1920x1080 resolution at 30 FPS. Tracking consistency and fatigue classification were visually assessed. Bounding boxes remained stable, speed calculations aligned with player movement, and fatigue trends appeared intuitive. The overlays were informative without interfering with game visibility.

# 10. Limitations and Future Work

While promising, the current implementation is limited to 2D motion and frame-based speed estimates. Future improvements could include:  
- Use of acceleration and direction changes for better fatigue modeling  
- Incorporating pose estimation for more nuanced analysis  
- Exporting fatigue stats to CSV or BI dashboards  
- Deploying as a live coaching tool during games  
- Applying the system to other sports (e.g., soccer, hockey)

# 11. Conclusion

This project demonstrates the feasibility of real-time fatigue tracking using only video input. By combining YOLOv8 object detection, Deep SORT tracking, and motion-based fatigue modeling, it delivers an innovative and practical solution for sports analytics. The system's non-invasive nature and visual clarity make it suitable for coaching, performance optimization, and injury prevention.

# 12. References

- Gabbett, T. J. (2016). The training—injury prevention paradox: should athletes be training smarter and harder? \*British Journal of Sports Medicine\*, 50(5), 273–280.  
- Pons, P., García, E., & Villalonga, C. (2020). Vision-based workload and fatigue estimation in team sports: A review. \*Computer Vision in Sports\*, Springer.  
- Ultralytics YOLOv8 Documentation: https://docs.ultralytics.com  
- Deep SORT GitHub: https://github.com/mikel-brostrom/Yolov5\_DeepSort\_Pytorch  
- OpenCV: https://opencv.org  
- Roboflow: https://roboflow.com