# A picture containing drawing Description automatically generatedSoftware Engineer Internship, Data Solutions

## Technical Project

### 1) System Design

I propose an end-to-end system that uses computer vision, machine learning, and a data-driven dashboard to analyze and evaluate player performance during practice sessions, providing coaches with both live feedback and detailed post-practice insights.

The system begins with high-definition cameras strategically installed around the practice facility to record synchronized video streams from multiple angles, capturing drills, scrimmages, and individual player movements. These video streams are sent to a local edge server equipped with GPUs, enabling real-time processing of player movements, poses, and ball trajectories with minimal latency.

The edge server serves as the first point of data ingestion, pre-processing the video and generating structured metrics and event data before sending them to a cloud storage solution (AWS S3/GCP Storage) for aggregation, storage, and further analysis. Streaming services like AWS Kinesis ensure continuous data flows, preventing any activity from being missed. This hybrid approach provides instant feedback during practice while maintaining scalable storage and historical analytics in the cloud.

Video data is processed using computer vision and AI technologies, including OpenCV for video preprocessing, YOLOv8/Detectron2 for player and ball detection, and Mediapipe/HRNet for pose estimation. DeepSORT is used for tracking players across frames. Processed data (player coordinates, movement features, event labels) are transmitted to a central cloud service where further analysis and aggregation occur. Scalable computation is supported using tools like Apache Spark/Flink, while APIs built with FastAPI serve metrics and queries to the frontend dashboard.

The system collects a range of key performance data points: player position and movement are captured as 2D coordinates from each camera feed, fused into 3D trajectories via calibration and triangulation; speed, acceleration, and distance are derived from frame-to-frame positional changes; shooting metrics such as release timing, trajectory arc, and success rate are extracted for per-player shot efficiency analysis; spatial proximity and movement patterns reveal interactions, passes, screens, defensive coverage, and overall team tactics; workload metrics summarize total distance covered, average speed, and time spent in high-intensity drills. Additionally, heart-rate monitors or accelerometers provide physiological data synchronized with video timestamps to enrich player workload and recovery analysis.

Coaches access an interactive dashboard that connects real-time insights with historical analysis. During practice, it displays live overlays of player positions, movement metrics, and performance highlights from the edge server, enabling immediate tactical adjustments. Structured metrics are stored in a time-series or analytical database like PostgreSQL or TimescaleDB, while raw video is archived in a cloud-based data lake. After practice, the same dashboard allows coaches to review sessions, link video clips to metrics, visualize trends, and identify areas for improvement. Built with React/Streamlit and visualized using Plotly/Chart.js, the dashboard ensures performance data is both instantly actionable and persistently accessible, while Docker/Kubernetes and Prometheus/Grafana maintain scalability, reliability, and system health.

Additional resources requested include GPU-equipped edge servers, a small team for initial video annotation and model validation, and wearable sensors for enhanced performance tracking. By combining automated data extraction with intuitive analytics, this system transforms raw practice footage into actionable, data-driven insights to support player development and coaching decisions.

A diagram of software components

Description automatically generated

Figure : High-level flowchart of NBA player performance analysis system.