Capstone 2025-2026

Sponsor: Prof. Raffaele de Amicis

XRDrone: real-time drone-to-VR video + vision effects

Project overview

Build a Unity application (Android/Quest 2 target) that receives a live video stream from a DJI Neo drone and renders it inside VR headset. The app must run real-time feature/object recognition on the incoming frames and display context-aware visual effects/overlays (e.g., bounding boxes, labels, particle cues, shader highlights) on top of the video in 3D space.

Core deliverables (MVP)

- Drone video to Quest 2
  - Receive live video (≥720p, ≥24 FPS) on Quest 2 and render it to a cylindrical/hemispherical surface in Unity (to reduce distortion).
  - End-to-end latency glass-to-glass target ≤300 ms (measure and report).
- On-device vision
  - Run real-time detection of at least two feature/object classes (e.g., “tree”, “vehicle”, “person”, “trail marker”, “QR/AprilTag”).
  - Attach overlays to detections (labels, bounding boxes, depth-aware marker, or stylized highlight via shader).
- UX inside VR
  - Simple cockpit HUD: stream status (bitrate/FPS/latency), toggle vision on/off, switch overlays, recenter view.
  - Safety interlocks: vision overlay clearly distinguishes detections vs guesses (confidence readout).
- Engineering docs
  - Architecture diagram, streaming protocol choice & rationale, model choice & benchmarks, measurement method, and known limits.

Acceptance criteria

- Streaming reliability (25%): Sustained ≥24 FPS with <2% dropped frames over 3-minute demo; clean recovery from a forced disconnect.
- Latency (15%): Measured ≤300 ms median glass-to-glass with documented method & tuning rationale.
- Vision accuracy & speed (20%): ≥0.5 F1 on a labeled mini-set you collect (≥150 frames), ≥15 FPS inference at runtime.
- VR UX & effects (20%): Readable overlays, unobtrusive HUD, at least one tasteful shader/VFX effect tied to detections.
- Engineering quality (20%): Clear architecture, reproducible build, config files, meaningful logs/metrics, limitations & ethical/safety notes.

Note

- While Unity is the recommended environment for rapid prototyping and VR integration, Unity is not a strict requirement, students may choose another engine or framework if they can meet the same deliverables and performance targets.