Today, 14 November 2025, our goal was to improve the real-time performance and stability of the live camera module within the Eyeora AI-CCTV system. The initial motivation for this work came from the extremely low frame rate the system was producing, approximately 0.5 FPS, which made the live feed unusable for practical surveillance or analytics. Our aim for the day was to achieve a smooth, stable video stream with real-time person and object detection, ideally reaching at least 25–30 FPS on CPU-only hardware. In addition to performance enhancement, we also set the objective of resolving any outstanding errors related to YOLO model initialization, improving stability across devices, and updating the architecture to better balance computational load.

At the start of the debugging process, we encountered a major limitation: the system continuously raised a runtime error indicating that "Torch is not compiled with CUDA enabled." This error occurred because the detection engine attempted to load the YOLO model on CUDA by default, even though the machine running the software did not have a CUDA-supported GPU. As a result, the live-camera pipeline failed during initialization, preventing any further processing. This limitation stemmed from hard-coded device selection logic that did not properly fall back to CPU mode when a GPU was unavailable. To fix this, we implemented a robust hardware detection routine that checks whether CUDA is available and automatically switches to CPU execution when required. This prevented the system from crashing during model initialization and guaranteed stability across different hardware platforms.

After resolving the initialization error, we focused on addressing the core challenge: extremely low FPS during live video processing. Several performance bottlenecks were identified. The system was running a heavyweight object detection model, performing pose estimation on every frame, rendering multiple overlays, executing clothing analysis continuously, and processing high-resolution frames. These combined tasks overwhelmed the CPU and drastically reduced throughput. To deal with these limitations, we implemented multiple optimisation strategies. First, we switched to the lightweight `yolo11n` model, which is significantly faster while still offering acceptable detection accuracy. Second, we reduced the input resolution of the camera feed to 480×360, which dramatically decreased the number of pixels per frame and therefore the computation time. Third, we restricted the object detection pipeline to a small set of COCO classes relevant to the system (persons, bottles, knives, laptops, mobile phones) rather than processing all 80 classes. Fourth, we implemented frame-skipping mechanisms so that pose estimation runs only every third frame and clothing classification only every fifth frame. This ensured that computationally expensive modules contribute meaningful information without overwhelming the system. Finally, we simplified on-screen drawing (bounding boxes, text, skeletons), reducing overlay complexity and increasing rendering speed.

Through these optimisation steps, the live camera pipeline experienced a significant improvement in performance and responsiveness. The system is now capable of running smoothly on CPU-only machines without errors, and the live feed delivers substantially increased frame rates compared to the original 0.5 FPS. The updated detection architecture is more efficient, stable, and scalable, while retaining the core features such as multi-object detection, person tracking, pose estimation, clothing analysis, and activity recognition. The fixes applied today have not only solved several critical issues but have also positioned the system for further enhancement in future updates.

The outcome of today's work is a fully operational, high-performance live camera system that runs reliably in real time even without GPU support. The next steps will involve testing the updated system on CUDA-enabled hardware for additional acceleration, refining multithreading to push frame rates beyond 30 FPS, and gradually reintroducing more advanced analytics modules while maintaining the improved speed. Overall, today's improvements mark a

major step forward in making the Eyeora AI-CCTV system production-ready and scalable for real-world deployment.