

AI Camera Tracking System – Executive Summary

Batra, Utkrisht
015806713

Dr. Khoshnoud
Electromechanical Engineering Technology
4/26/2026

1. How long did it take to complete this executive summary? Approximately 5 hours
2. Did you attend or watch the lesson provided by Mr. Paul Hottinger regarding Information Literacy (check the correct box)? ☐ YES ☒ NO

AI Camera Tracking System – Executive Summary

The objective of this senior project was to design, build, and verify an offline AI camera tracking system for BillyBOT, a campus guide robot platform. The main issue this project addresses is the weakness of camera systems that rely heavily on internet access, cloud processing, or off-site storage. Those systems can be useful for normal home security, but they are not ideal for a mobile robot that needs direct access to its own vision data. For BillyBOT, the camera system needs to detect a person, move a pan-tilt head, save useful data locally, and keep working even when internet access is unavailable.

The final design is a modular tracking system built around an NVIDIA Jetson Orin Nano, an Arducam IMX477 HQ camera, a Yahboom 2-degree-of-freedom metal pan-tilt assembly, DS3218MG-class servos, a PCA9685 pulse-width modulation driver, and local storage through microSD and NVMe storage. The Jetson Orin Nano was selected because it is a small, embedded computing platform with an NVIDIA Ampere GPU, 1,024 CUDA cores, 32 Tensor Cores, a 6-core Arm CPU, 102 GB/s memory bandwidth, and external NVMe support [1]. Those features make it a practical choice for a student-built local vision system instead of a cloud-dependent camera. The tracking software uses Python and OpenCV-based detection logic to identify a human target and calculate the target's position relative to the center of the image frame. OpenCV's DNN module supports loading serialized neural network models from different frameworks, which made it useful for local detection and inference testing [2]. Once the target position is found, the software converts the pixel error into pan and tilt commands.

The system intentionally limits automated motion to a safe software envelope of pan 60° to 120° and tilt 30° to 130°. The physical pan-tilt hardware can move farther than this, but using the full mechanical range during automated tracking could risk pulling wires, stressing the camera mount, or forcing the assembly into awkward positions. For that reason, the project treats the software limits as part of the safety design, not just a programming detail.

The external research supported the decision to build an offline system instead of copying a normal consumer camera setup. Commercial systems such as Ring and Wyze offer useful features, but recorded video access, cloud storage, and smart detection features are commonly tied to subscription plans [3], [4]. That may be acceptable for a home camera, but it does not match the needs of a robot that should process its own data and remain useful without cloud access. Wireless security systems also have a known vulnerability against jamming. The Federal Communications Commission states that jammers can prevent Wi-Fi-enabled devices from connecting to the internet and that jammer use is prohibited in the United States except under specific federal authorization [5]. A more recent FCC statement also notes that bad actors have used Wi-Fi jammers to disrupt security systems, security cameras, and cellphones during unauthorized entries and burglaries [6]. Based on these sources, my position is that BillyBOT benefits more from a local, self-contained tracking module than from a cloud-connected or fixed camera approach.

The design process started with the main functional requirements: local person detection, pan-tilt actuation, local evidence and log storage, software motion limits, and a path for future BillyBOT integration. Hardware was selected based on processing capability, cost, serviceability, and how

easily the parts could be integrated by a student team. The Arducam IMX477 HQ camera was selected because it uses the Sony IMX477 image sensor and has Jetson-compatible documentation for embedded camera use [7]. The CSI-2 camera connection was selected because MIPI CSI-2 is a high-speed interface for transmitting still and video images from image sensors to application processors [8]. For servo control, the PCA9685 was selected because it is an I2C-controlled 16-channel PWM device with 12-bit output resolution [9]. The I2C bus was appropriate for this control path because it is a bidirectional two-wire inter-IC bus intended for efficient communication between integrated circuits [10].

The mechanical design focused on making the camera mount simple, protected, and easy to revise. The only fully custom mechanical part in the system is the camera mount and protective cover that holds the IMX477 camera to the pan-tilt assembly. Additive manufacturing was used because it allowed quick changes to hole placement, cable clearance, and overall fitment. This mattered because the camera and pan-tilt head had to be compact enough for BillyBOT while still being easy to remove, inspect, and improve. Several prototype revisions were made before the final physical arrangement shown in Appendix A.

The electrical and control design was kept modular so that each major part of the system had a clear role. The IMX477 camera connects to the Jetson Orin Nano through the CSI-2 camera interface. The PCA9685 connects to the Jetson through I2C and produces the PWM signals used to command the pan and tilt servos. The system uses regulated DC power sized for the Jetson processing load and the servo demand. Local storage is handled with a 64 GB microSD card and a 256 GB NVMe solid-state drive. This gives the system removable setup storage as well as faster onboard storage for logging and future data capture.

The tracking software was organized around a state-machine structure instead of only reacting frame by frame. The tracker begins with an idle or centered condition, then enters target acquisition when a person is detected. Once a target is selected, the system enters a tracking state and tries to keep the person near the center of the frame. If the target is briefly blocked or lost, the logic can hold, attempt local reacquisition, or return to center. This approach made the behavior easier to understand and safer to expand later. It also gives future students a cleaner structure for adding better multi-person handling, smoother motion control, and ROS 2 integration.

The current verification used automated baseline testing, modeled simulation support, and integrated observation with BillyBOT. In the baseline validation run, the system started successfully, opened the camera, and loaded the detector. The sampled run processed 10 frames, recorded detections in 10 out of 10 frames, made 9 face-based target selections, recorded 0 pan clamp hits, 0 tilt clamp hits, and 0 target losses. The same run reported an average processing rate of 3.11 frames per second, as shown in Table 1 in Appendix A. This result is useful because it proves the basic detection and tracking path worked under controlled conditions. However, it should not be treated as final optimized performance because the test sample was small and came from a dry-run baseline. As shown in Appendix B, the detection-frame success rate for this sampled run was 100%.

Simulation was also used to better understand the tracker's behavior before treating the system as fully validated. One modeled plot showed the relationship between target horizontal position and commanded pan response. That model used a 1280-pixel frame width, a 640-pixel center position, a 28-pixel pan deadband, a pan gain of 24.5, a maximum pan step of 3.2° , and a 90° pan center. This helped show how pixel error turns into commanded pan motion. A separate safety-boundary plot helped explain how the software limits keep commanded motion inside the allowed range. These simulation outputs are useful for explaining the logic of the design, but they are not closed-loop hardware proof.

The project successfully moved from concept to a functioning offline AI camera tracking prototype. The system can run local detection, command a pan-tilt assembly, save data locally, and stay inside defined software motion limits. The estimated prototype cost was approximately \$504.44, including the Jetson Orin Nano development kit, IMX477 camera, pan-tilt assembly, PCA9685 driver, power conversion hardware, IR illuminator, storage, wiring, case material, and fasteners. This supports the goal of building a relatively affordable student development platform rather than depending on a closed commercial system.

The project still has limitations that should be stated clearly. The current servo timing information represents software command dispatch and update timing only. It should not be interpreted as true physical servo settling time because the system does not have encoder feedback. The measured 3.11 FPS value is also not the final optimized inference rate. Additional structured testing is still needed for multi-person target selection, faster target motion, low-light operation, formal end-to-end latency, and target-centering accuracy. These limits do not make the prototype unsuccessful, but they define what still needs to be proven before the system can be called fully validated.

The conclusion is that the project produced a credible and expandable offline AI camera tracking system for BillyBOT. The strongest parts of the design are the local processing architecture, modular hardware stack, repaired and verified camera path, software-safe pan and tilt limits, and repeatable logging workflow. The project directly addresses the original problem by reducing cloud dependence and making the camera system more suitable for robotic integration. Based on the current evidence, the design should be described as functionally demonstrated and ready for further validation, not as production-ready. The recommended next steps are to complete structured human-in-the-loop testing, improve false-positive rejection, test more multi-person scenarios, verify low-light operation, and measure true end-to-end latency and target-centering accuracy. Future work should also include cleaner cable routing, more polished BillyBOT mounting, and possible servo feedback if more accurate motion verification is required.

Work Cited

- [1] NVIDIA, “Jetson Orin Nano Super Developer Kit,” NVIDIA. Accessed: Apr. 24, 2026. Available: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/>
- [2] OpenCV, “Deep Neural Network module,” OpenCV Documentation. Accessed: Apr. 24, 2026. Available: https://docs.opencv.org/4.x/d6/d0f/group__dnn.html
- [3] Ring, “Understanding Ring Protect subscriptions,” Ring Support. Accessed: Apr. 24, 2026. Available: <https://ring.com/support/articles/lw54x/General-Information-on-Ring-Protect-Subscription-Plans>
- [4] Wyze Labs, “Home Monitoring & Security App - Wyze Service Plans,” Wyze. Accessed: Apr. 24, 2026. Available: <https://www.wyze.com/pages/service-plans>
- [5] Federal Communications Commission, “Warning: Jammer Use is Prohibited,” Enforcement Advisory No. 2014-05, Dec. 8, 2014. Accessed: Apr. 24, 2026. Available: <https://docs.fcc.gov/public/attachments/da-14-1785a1.pdf>
- [6] Federal Communications Commission, “Statement of Commissioner Anna M. Gomez Re: Promoting Technological Solutions to Combat Contraband Wireless Device Use in Correctional Facilities,” FCC 25-65, Sept. 30, 2025. Accessed: Apr. 24, 2026. Available: <https://docs.fcc.gov/public/attachments/FCC-25-65A3.pdf>
- [7] Arducam, “12MP IMX477,” Arducam Documentation. Accessed: Apr. 24, 2026. Available: <https://docs.arducam.com/Nvidia-Jetson-Camera/Native-Camera/imx477/>
- [8] MIPI Alliance, “Camera Serial Interface 2 (MIPI CSI-2),” MIPI Alliance. Accessed: Apr. 24, 2026. Available: <https://www.mipi.org/specifications/csi-2>
- [9] NXP Semiconductors, “PCA9685: 16-channel, 12-bit PWM Fm+ I2C-bus LED driver,” NXP Semiconductors. Accessed: Apr. 24, 2026. Available: <https://www.nxp.com/products/power-drivers/lighting-driver-and-controller-ics/led-drivers/16-channel-12-bit-pwm-fm-plus-ic-bus-led-driver:PCA9685>
- [10] NXP Semiconductors, “UM10204: I2C-bus specification and user manual, Rev. 7.0,” Oct. 1, 2021. Accessed: Apr. 24, 2026. Available: https://www.nxp.com/documents/user_manual/UM10204.pdf

Appendix A: Tables and Figures



Figure 1. AI camera tracking system block diagram.



Figure 2. Final AI camera tracking prototype prepared for BillyBOT integration.

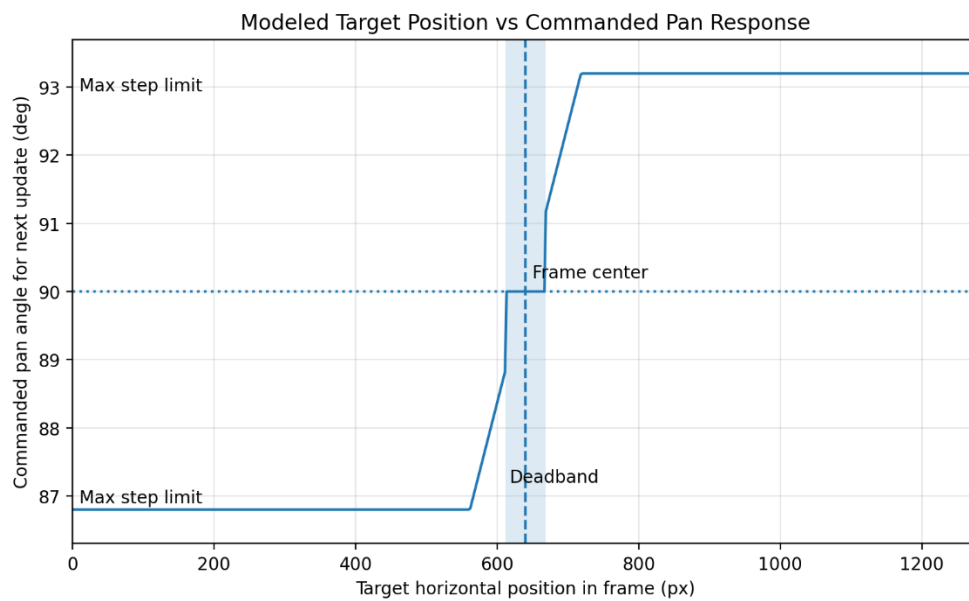


Figure 3. Modeled target position versus commanded pan response.

Table 1. Baseline Validation Dry-Run Metrics

Metric	Result
Startup success	Yes
Camera opened	Yes
Detector loaded	Yes
Average FPS	3.11
Frames processed	10
Frames with detections	10
Face-based target selections	9
Pan clamp hits	0
Tilt clamp hits	0
Target losses	0
Successful reacquisitions	0
Target switches	0

Table 2. Design Requirements Verification Status

Requirement	Current Verification Status
Offline/onboard processing	Verified in baseline run
Local data storage	Verified in baseline run
Pan-tilt actuation response	Partially verified / Improved with command-path evidence
Human target tracking	Partially verified with automated baseline and working live-follow capability
Low-light operation	Pending live test
Quantitative latency and accuracy	Partially improved, limited by pending structured live testing and lack of servo encoder feedback

Appendix B: Sample Calculation

Sample Calculation: Detection-Frame Success Rate

Given:

Frames processed = 10

Frames with detections = 10

Equation:

Detection-frame success rate = $(\text{Frames with detections} / \text{Frames processed}) \times 100\%$

Substitution:

Detection-frame success rate = $(10 / 10) \times 100\%$

Result:

Detection-frame success rate = 100%

Interpretation:

For the sampled baseline dry-run sequence, the tracker recorded detections in every processed frame. This supports basic detector functionality under controlled test conditions. However, this result should not be treated as full system validation because the sample size was limited and broader structured live testing is still required.