HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN I**

**⎯⎯⎯⎯⎯⎯⎯⎯⎯**



**BÁO CÁO GIỮA KÌ**

**MÔN HỌC: IOT VÀ ỨNG DỤNG**

**Nhóm lớp học: Nhóm 05**

**Nhóm bài tập lớn: Nhóm 13**

**Đề tài: Xe trông nhà thông minh**

**Danh sách sinh viên:**

**Họ và tên Mã sinh viên**

**Nguyễn Mạnh Cường B22DCCN100**

**Trần Đức Phương B22DCCN640**

**Đỗ Thành Đạt B21DCCN184**

**Bùi Văn Hiến B22DCCN292**

**Giảng viên giảng dạy: Kim Ngọc Bách**

**HÀ NỘI – 2025**

Nội dung

[A. Xác định mục tiêu và phạm vi hệ thống 5](#_Toc211437278)

[I.Mục tiêu hệ thống 5](#_Toc211437279)

[1. Vấn đề thực tế 5](#_Toc211437280)

[2. Mục tiêu IoT 5](#_Toc211437281)

[3. Kỳ vọng 6](#_Toc211437282)

[II.Phạm vi triển khai 6](#_Toc211437283)

[1.Số lượng thiết bị: 6](#_Toc211437284)

[2.Môi trường hoạt động: 6](#_Toc211437285)

[3.Khu vực thử nghiệm: 6](#_Toc211437286)

[III.Tiêu chí thành công 6](#_Toc211437287)

[1.Độ chính xác: 6](#_Toc211437288)

[2.Độ trễ: 7](#_Toc211437289)

[3.Độ tin cậy: 7](#_Toc211437290)

[4.Hiệu suất hệ thống: 7](#_Toc211437291)

[5.Chi phí: 7](#_Toc211437292)

[6.Khả năng mở rộng: 7](#_Toc211437293)

[IV.Kết quả mong đợi 7](#_Toc211437294)

[B. Thu thập yêu cầu từ các bên liên quan 8](#_Toc211437295)

[I. Bối cảnh vấn đề 8](#_Toc211437296)

[II. Yêu cầu từ các bên liên quan 8](#_Toc211437297)

[1. Người dùng cuối (Nhân viên bảo vệ & Quản lý an ninh) 8](#_Toc211437298)

[2. Quản lý kỹ thuật (Đội ngũ phát triển & Bảo trì) 9](#_Toc211437299)

[3. Ban lãnh đạo (Giám đốc & Quản lý cấp cao) 9](#_Toc211437300)

[III. Phương pháp thu thập yêu cầu 10](#_Toc211437301)

[1.Phỏng vấn: 10](#_Toc211437302)

[2.Khảo sát: 10](#_Toc211437303)

[3.Quan sát thực tế: 10](#_Toc211437304)

[4.Nghiên cứu tài liệu: 10](#_Toc211437305)

[5.Thử nghiệm pilot: 10](#_Toc211437306)

[C. Xác định yêu cầu chức năng 11](#_Toc211437307)

[I. Các chức năng cần có của hệ thống 11](#_Toc211437308)

[1.Thu thập dữ liệu: 11](#_Toc211437309)

[2.Truyền dữ liệu: 11](#_Toc211437310)

[3.Xử lý và phân tích dữ liệu: 11](#_Toc211437311)

[4.Điều khiển/Ra lệnh: 11](#_Toc211437312)

[5.Cảnh báo: 11](#_Toc211437313)

[II.Đặc tả công việc 12](#_Toc211437314)

[D.Xác định yêu cầu phi chức năng 12](#_Toc211437315)

[I.Các yêu cầu phi chức năng 12](#_Toc211437316)

[1. Hiệu năng 12](#_Toc211437317)

[2. Bảo mật 13](#_Toc211437318)

[3. Độ tin cậy 13](#_Toc211437319)

[4. Khả năng mở rộng 13](#_Toc211437320)

[5. Chi phí và Năng lượng 14](#_Toc211437321)

[7. Khả năng bảo trì 14](#_Toc211437322)

[E. Ràng buộc pháp lý 14](#_Toc211437323)

[F. Phân tích ràng buộc kỹ thuật và môi trường 15](#_Toc211437324)

[I. Môi trường hoạt động 15](#_Toc211437325)

[II. Ràng buộc pháp lý 16](#_Toc211437326)

[III. Tài nguyên thiết bị 16](#_Toc211437327)

[G. CÁC LÝ THUYẾT VÀ CÔNG NGHỆ ÁP DỤNG 17](#_Toc211437328)

[I. Kiến trúc IoT 17](#_Toc211437329)

[1. Mô hình ba tầng (Three-tier Architecture) 17](#_Toc211437330)

[2. Điện toán biên (Edge Computing) 17](#_Toc211437331)

[II 17](#_Toc211437332)

[1. Nhận diện đối tượng bằng YOLO (You Only Look Once) 17](#_Toc211437333)

[2. Thị giác máy tính với OpenCV 18](#_Toc211437334)

[3. Xử lý ảnh trên ESP32-CAM 18](#_Toc211437335)

[III. Hệ thống nhúng và Vi điều khiển 18](#_Toc211437336)

[IV. Giao thức truyền thông 18](#_Toc211437337)

[1. Kết nối WiFi 802.11n (2.4GHz): 18](#_Toc211437338)

[2. Giao thức WebSocket: 19](#_Toc211437339)

[3. Bluetooth Classic (BR/EDR): 19](#_Toc211437340)

[4. Định dạng dữ liệu JSON: 19](#_Toc211437341)

[V. Cảm biến và Điều khiển 19](#_Toc211437342)

[VI. Thuật toán và Xử lý 20](#_Toc211437343)

[1. Thuật toán tránh vật cản: 20](#_Toc211437344)

[2. Bộ lọc Kalman: 20](#_Toc211437345)

[VII. Cập nhật Firmware OTA 20](#_Toc211437346)

[VIII. Bảo mật 20](#_Toc211437347)

[IX. Công nghệ phát triển ứng dụng 21](#_Toc211437348)

[1. Python với OpenCV và YOLO: 21](#_Toc211437349)

[2. Lập trình ESP32 với Arduino Framework: 21](#_Toc211437350)

[3. Ứng dụng di động: 21](#_Toc211437351)

[4. Thư viện giao tiếp: 21](#_Toc211437352)

# A. Xác định mục tiêu và phạm vi hệ thống

## I.Mục tiêu hệ thống

### 1. Vấn đề thực tế

Hiện nay, công tác giám sát và tuần tra an ninh gặp nhiều hạn chế:

* Chi phí nhân lực cao: Cần nhiều nhân viên bảo vệ làm việc 24/7 để giám sát các khu vực quan trọng.
* Hạn chế con người: Nhân viên có thể mệt mỏi, mất tập trung, không quan sát được toàn diện.
* Nguy hiểm: Tuần tra các khu vực nguy hiểm (nhà máy, kho hóa chất) gây rủi ro cho nhân viên.
* Phát hiện chậm: Camera truyền thống chỉ ghi hình, không tự động phát hiện và cảnh báo xâm nhập.
* Thiếu linh hoạt: Camera cố định không thể di chuyển để theo dõi hoặc khảo sát nhiều khu vực.

Do đó, cần một giải pháp giám sát thông minh, tự động, có khả năng di chuyển và phát hiện xâm nhập kịp thời.

### 2. Mục tiêu IoT

* Giám sát tự động: Thu thập và theo dõi các thông số quan trọng như khoảng cách đến vật cản, mức pin, tốc độ di chuyển, và truyền video real-time về laptop.
* Điều khiển từ xa đa kênh: Điều khiển xe qua Bluetooth (smartphone) cho chế độ Manual, đồng thời nhận lệnh cảnh báo từ laptop qua WiFi khi AI phát hiện người.
* Navigation tự động thông minh: Tự động tránh vật cản dựa trên dữ liệu cảm biến siêu âm, điều chỉnh tốc độ và hướng di chuyển phù hợp với môi trường.
* Nhận diện và cảnh báo AI: Sử dụng YOLO trên laptop để nhận diện người trong video stream, tự động kích hoạt cảnh báo âm thanh trên cả laptop và xe khi phát hiện xâm nhập.
* Cập nhật firmware từ xa: Hỗ trợ OTA (Over-The-Air) update qua WiFi, cho phép nâng cấp và sửa lỗi mà không cần can thiệp vật lý.

### 3. Kỳ vọng

* Triển khai tại kho hàng, văn phòng, khu công nghiệp
* Giảm 30-50% chi phí nhân lực giám sát
* Phát hiện và cảnh báo xâm nhập kịp thời
* Giám sát từ xa các khu vực nguy hiểm

## II.Phạm vi triển khai

### 1.Số lượng thiết bị:

* 1 xe robot tự hành (prototype) tích hợp ESP32-CAM, động cơ DC, cảm biến siêu âm, và module cảnh báo.
* 1 laptop xử lý AI (YOLO) để nhận diện người từ video stream.
* 1 smartphone (Android/iOS) để điều khiển xe qua Bluetooth.
* 1 router WiFi hoặc laptop hotspot để kết nối ESP32-CAM với laptop.

### 2.Môi trường hoạt động:

* Ngoài trời (outdoor) → điều kiện thời tiết bình thường, nhiệt độ 20-35°C, không mưa, ánh sáng tự nhiên ban ngày.
* Yêu cầu: Mặt đất tương đối phẳng (sân, vỉa hè, sân trường), không có vật cản phức tạp, khoảng cách WiFi <15m để đảm bảo streaming ổn định.
* Nguồn điện: Pin Li-Po 7.4V (2200mAh) cho phép hoạt động liên tục 1-2 giờ.
* Lưu ý: Các linh kiện điện tử cần được bảo vệ khỏi bụi bẩn và nắng trực tiếp, tránh nhiệt độ quá cao ảnh hưởng đến ESP32-CAM và pin.

### 3.Khu vực thử nghiệm:

* Diện tích: 50-100m² (sân trường, sân nhà, bãi đỗ xe).
* Điều kiện: Có nguồn điện để sạc pin, WiFi ổn định hoặc sử dụng laptop hotspot di động, đủ không gian cho xe di chuyển và thử nghiệm chế độ tự động tránh vật cản.

## III.Tiêu chí thành công

### 1.Độ chính xác:

* Độ chính xác nhận diện người (AI) > 85%.
* Sai số đo khoảng cách cảm biến siêu âm < ±2cm.
* Tỷ lệ false positive (cảnh báo nhầm) < 10%.

### 2.Độ trễ:

* Độ trễ video streaming < 1s.
* Độ trễ điều khiển Bluetooth (Manual mode) < 200ms.
* Thời gian phản ứng cảnh báo khi phát hiện người < 1s.

### 3.Độ tin cậy:

* Tỷ lệ truyền video thành công > 95%.
* Xe tránh vật cản thành công 100%, không va chạm trong chế độ Auto.
* Kết nối WiFi/Bluetooth ổn định, tự động reconnect khi mất kết nối.

### 4.Hiệu suất hệ thống:

* FPS video streaming: 10-15 frames/giây.
* FPS xử lý AI trên laptop > 10 frames/giây.
* Thời gian hoạt động liên tục: 1-2 giờ với 1 lần sạc pin.

### 5.Chi phí:

* Tổng chi phí đầu tư ban đầu cho 1 xe prototype < 2 triệu VNĐ (bao gồm ESP32-CAM, động cơ, cảm biến, khung xe, pin).

### 6.Khả năng mở rộng:

* Kiến trúc sẵn sàng mở rộng thành đội robot tuần tra (multi-robot system) trong tương lai.

## IV.Kết quả mong đợi

* Tự động hóa giám sát an ninh → giảm 30-50% chi phí nhân lực bảo vệ, tăng hiệu quả tuần tra 24/7.
* Phát hiện xâm nhập thông minh → cảnh báo kịp thời khi phát hiện người trong khu vực cấm, ghi lại bằng chứng video real-time.
* Điều khiển linh hoạt từ xa → người vận hành có thể giám sát và điều khiển xe qua smartphone, theo dõi trạng thái (pin, khoảng cách, chế độ hoạt động) mọi lúc mọi nơi.
* Nền tảng nghiên cứu IoT & AI → dữ liệu video thu thập liên tục giúp cải thiện thuật toán AI, phát triển thêm tính năng nhận diện khuôn mặt, theo dõi đối tượng, phân tích hành vi.
* Giải pháp an toàn → giám sát các khu vực nguy hiểm (nhà máy, kho hóa chất) mà không cần con người tiếp cận trực tiếp.
* Khả năng mở rộng → làm nền tảng để phát triển hệ thống đội robot tuần tra tự động, tích hợp với hệ thống báo động và camera giám sát hiện có.

# B. Thu thập yêu cầu từ các bên liên quan

## I. Bối cảnh vấn đề

* Các khu vực cần giám sát an ninh như kho hàng, nhà máy, văn phòng thường có diện tích rộng với nhiều góc khuất.
* Hiện nay, việc giám sát chủ yếu dựa vào nhân viên bảo vệ tuần tra thủ công hoặc camera cố định → dễ bỏ sót khu vực, phát hiện xâm nhập chậm trễ, chi phí nhân lực cao, nguy hiểm khi tuần tra khu vực nguy hiểm.
* Hệ thống camera giám sát truyền thống chỉ ghi hình thụ động, không có khả năng di chuyển linh hoạt hay phát hiện và cảnh báo tự động khi có người xâm nhập trái phép.
* Ban quản lý muốn ứng dụng IoT kết hợp AI để xây dựng robot tuần tra tự động, có khả năng nhận diện người xâm nhập, cảnh báo kịp thời và giảm thiểu rủi ro cho nhân viên bảo vệ.

## II. Yêu cầu từ các bên liên quan

### 1. Người dùng cuối (Nhân viên bảo vệ & Quản lý an ninh)

Hiển thị:

* Muốn xem video streaming real-time từ xe trên laptop hoặc smartphone để giám sát khu vực.
* Xem số lượng người phát hiện, vị trí người (bounding box), confidence score của AI.

Cảnh báo:

* Nhận thông báo tức thì (âm thanh + visual alert) khi AI phát hiện người xâm nhập trong khu vực giám sát.
* Cảnh báo khi xe gặp sự cố (mất kết nối WiFi/Bluetooth, vật cản chặn đường quá lâu).

Điều khiển:

* Có quyền điều khiển xe từ xa qua smartphone (chế độ Manual): tiến/lùi/rẽ trái/rẽ phải.
* Có nút Emergency Stop để dừng xe khẩn cấp từ xa khi phát hiện rủi ro.
* Chuyển đổi chế độ Manual/Auto linh hoạt theo tình huống.

### 2. Quản lý kỹ thuật (Đội ngũ phát triển & Bảo trì)

Giám sát hệ thống:

* Theo dõi telemetry data (FPS video, FPS AI, độ trễ mạng, nhiệt độ ESP32-CAM) để đánh giá hiệu suất.

Bảo trì và nâng cấp:

* Hỗ trợ OTA firmware update để sửa lỗi và nâng cấp tính năng từ xa mà không cần thu hồi xe.
* Khả năng rollback firmware tự động nếu update thất bại để đảm bảo hệ thống luôn hoạt động.

Mở rộng:

* Kiến trúc cho phép thêm cảm biến mới (nhiệt độ, độ ẩm, GPS, gas sensor) mà không cần thay đổi firmware lớn.
* Hỗ trợ tích hợp với hệ thống báo động và camera giám sát hiện có.

### 3. Ban lãnh đạo (Giám đốc & Quản lý cấp cao)

Hiệu quả kinh tế:

* Giảm 30-50% chi phí nhân lực bảo vệ bằng cách sử dụng robot tuần tra tự động.
* Tăng độ phủ giám sát mà không cần đầu tư thêm camera cố định đắt tiền.

Báo cáo và phân tích:

* Dashboard tổng quan: tổng số lần phát hiện xâm nhập, thời gian hoạt động, tỷ lệ cảnh báo đúng/sai.
* Báo cáo định kỳ (ngày/tuần/tháng) về tình hình an ninh, xu hướng xâm nhập.

Khả năng mở rộng:

* Hệ thống có thể nhân rộng thành đội robot tuần tra (fleet management) trong tương lai.
* Nền tảng nghiên cứu để phát triển thêm tính năng AI nâng cao (nhận diện khuôn mặt, phân tích hành vi).

## III. Phương pháp thu thập yêu cầu

### 1.Phỏng vấn:

* Làm việc với nhân viên bảo vệ và quản lý an ninh để hiểu rõ khó khăn trong công tác tuần tra, giám sát (khu vực khó tiếp cận, thời gian tuần tra, tần suất phát hiện xâm nhập).
* Trao đổi với đội ngũ kỹ thuật về khả năng tích hợp với hệ thống camera và báo động hiện có.

### 2.Khảo sát:

* Gửi bảng hỏi cho ban quản lý để thu thập nhu cầu về tính năng (chế độ tự động/thủ công, loại cảnh báo, báo cáo an ninh).
* Thu thập ý kiến về ngân sách đầu tư, thời gian triển khai và yêu cầu đào tạo sử dụng.

### 3.Quan sát thực tế:

* Đội ngũ phát triển đến trực tiếp khu vực giám sát (kho hàng, sân trường, nhà máy) để:
  + Khảo sát địa hình (mặt đất phẳng/gồ ghề, vật cản cố định).
  + Kiểm tra phủ sóng WiFi, điểm mù tín hiệu.
  + Đánh giá điều kiện ánh sáng (ban ngày/ban đêm) ảnh hưởng đến camera và AI.
  + Xác định vị trí đặt trạm sạc pin và điểm xuất phát của xe.

### 4.Nghiên cứu tài liệu:

* Tham khảo các dự án robot tuần tra, xe tự hành tương tự để học hỏi kinh nghiệm.
* Nghiên cứu tiêu chuẩn an ninh và quy định pháp lý về giám sát tự động.

### 5.Thử nghiệm pilot:

* Triển khai prototype trong môi trường thực tế với quy mô nhỏ (1 xe, 1 khu vực).
* Thu thập feedback từ người dùng để cải tiến tính năng trước khi triển khai rộng.

# C. Xác định yêu cầu chức năng

## I. Các chức năng cần có của hệ thống

### 1.Thu thập dữ liệu:

* Thu thập video/hình ảnh real-time từ camera OV2640 trên ESP32-CAM.
* Đo khoảng cách đến vật cản liên tục từ cảm biến siêu âm HC-SR04.

### 2.Truyền dữ liệu:

* Streaming video về laptop qua WiFi (MJPEG format, 10-15 FPS).
* Nhận lệnh điều khiển từ smartphone qua Bluetooth.
* Nhận lệnh cảnh báo từ laptop qua WiFi.

### 3.Xử lý và phân tích dữ liệu:

* Xử lý AI trên laptop: nhận diện người từ video stream bằng YOLO.
* Xử lý logic tránh vật cản trên ESP32-CAM dựa trên dữ liệu cảm biến siêu âm.
* Phân tích và lọc dữ liệu: loại bỏ nhiễu cảm biến, áp dụng NMS (Non-Maximum Suppression) cho kết quả AI.

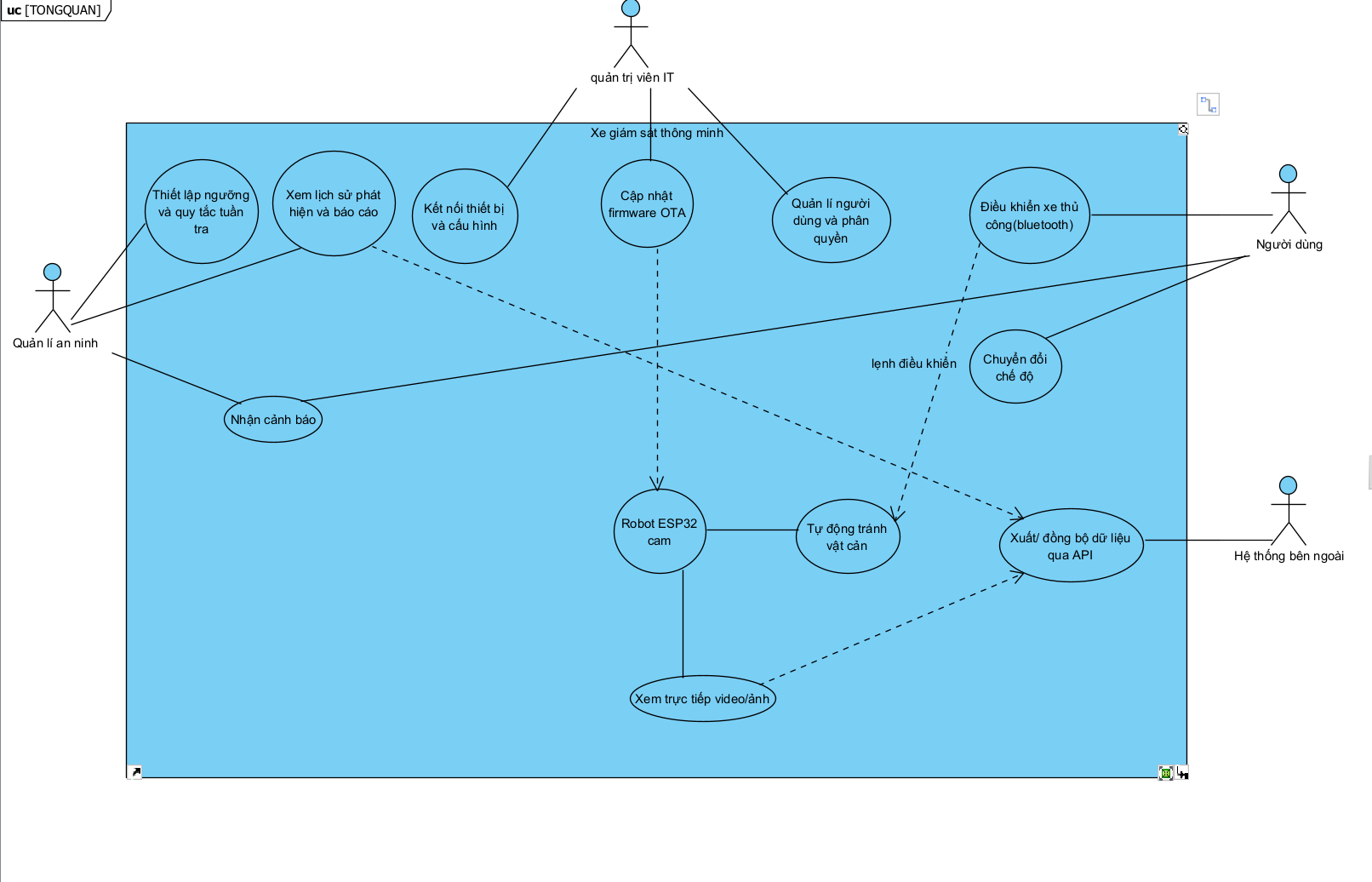
### 4.Điều khiển/Ra lệnh:

* Điều khiển di chuyển xe (tiến/lùi/trái/phải/dừng) từ smartphone qua Bluetooth.
* Chuyển đổi chế độ Manual/Auto từ app.
* Gửi lệnh Emergency Stop từ app hoặc laptop.
* Laptop gửi lệnh cảnh báo (bật/tắt buzzer) về ESP32-CAM khi phát hiện/mất người.
* Cập nhật firmware từ xa qua OTA.

### 5.Cảnh báo:

* Kích hoạt buzzer/loa trên xe khi nhận lệnh cảnh báo từ laptop.
* Hiển thị visual alert (màu đỏ, chữ "ALERT!") trên UI laptop/app.

## II.Đặc tả công việc



# D.Xác định yêu cầu phi chức năng

## I.Các yêu cầu phi chức năng

### 1. Hiệu năng

Độ trễ tối đa:

* Truyền video: độ trễ < 1s từ ESP32-CAM đến laptop
* Điều khiển Bluetooth: độ trễ < 200ms từ khi nhấn nút đến khi xe phản ứng
* Cảnh báo AI: thời gian từ phát hiện người đến kích hoạt còi < 500ms
* Cập nhật firmware OTA: tốc độ tải lên tối thiểu 50KB/s

Tần suất lấy mẫu:

* Camera: 10-15 khung hình/giây (FPS)
* Cảm biến siêu âm: đo khoảng cách mỗi 100ms (10 lần/giây)
* Dữ liệu giám sát: gửi mỗi 1-2 giây (pin, khoảng cách, tốc độ, chế độ)
* Xử lý AI: xử lý tối thiểu 10 khung hình/giây để đảm bảo phát hiện kịp thời

Khả năng chịu tải:

* Hệ thống phải xử lý được 1 luồng video HD (640x480) đồng thời với điều khiển và giám sát
* Laptop AI phải xử lý được YOLO với mức sử dụng CPU < 70%
* ESP32-CAM phải xử lý được đồng thời: truyền video, Bluetooth, WiFi, đọc cảm biến, điều khiển động cơ
* Băng thông WiFi tối thiểu: 2 Mbps cho truyền video ổn định

### 2. Bảo mật

Mã hóa:

* Kết nối WiFi: sử dụng mã hóa WPA2/WPA3
* Bluetooth: ghép nối bảo mật với mã PIN hoặc khóa truy cập
* Cập nhật firmware OTA: xác thực checksum (MD5/SHA256) trước khi cài đặt
* Truyền dữ liệu nhạy cảm: sử dụng HTTPS/TLS

### 3. Độ tin cậy

Cơ chế dự phòng:

* Dừng khẩn cấp tự động khi phát hiện vật cản < 20cm (không phụ thuộc WiFi/Bluetooth)
* Bộ đếm thời gian giám sát: tự động khởi động lại ESP32-CAM nếu firmware bị treo
* Giám sát pin: cảnh báo và dừng xe an toàn khi pin < 20%
* Khôi phục OTA: tự động quay lại firmware cũ nếu cập nhật thất bại hoặc khởi động không thành công

### 4. Khả năng mở rộng

Hỗ trợ thêm thiết bị mới:

* Kiến trúc module cho phép thêm cảm biến mới (nhiệt độ, độ ẩm, khí gas, GPS) chỉ bằng cách thêm module firmware
* Các chân GPIO dự phòng trên ESP32-CAM để kết nối thêm thiết bị
* Bus I2C/SPI có thể kết nối nhiều cảm biến cùng lúc
* Firmware hỗ trợ đăng ký cảm biến động (không cần biên dịch lại)

### 5. Chi phí và Năng lượng

Quản lý năng lượng:

* Pin Li-Po 7.4V 2200mAh: đủ cho 1-2 giờ hoạt động liên tục
* Chế độ ngủ khi không sử dụng: giảm 80% tiêu thụ điện
* Tự động giảm khung hình camera khi không cần thiết (từ 15 FPS xuống 5 FPS)
* Đèn LED báo hiệu tắt tự động sau 30 giây để tiết kiệm pin
* Chế độ tiết kiệm năng lượng: tắt truyền video WiFi khi chỉ cần điều khiển Bluetooth

6. Khả năng sử dụng

Dễ sử dụng:

* Ứng dụng di động: giao diện trực quan, dễ hiểu, thời gian học < 5 phút
* Giao diện laptop: hiển thị rõ ràng video, trạng thái, cảnh báo
* Kết nối một chạm: tự động tìm và kết nối ESP32-CAM
* Phản hồi trực quan: đèn LED, còi báo, màu sắc giao diện để người dùng biết trạng thái

### 7. Khả năng bảo trì

Dễ bảo trì:

* Mã nguồn có chú thích đầy đủ, tuân thủ quy ước lập trình
* Thiết kế module: mỗi module độc lập, dễ sửa lỗi
* Hệ thống nhật ký: ghi lại các sự kiện quan trọng để gỡ lỗi
* Cập nhật OTA: sửa lỗi và nâng cấp từ xa mà không cần thu hồi thiết bị

# E. Ràng buộc pháp lý

Quy định tần số vô tuyến:

* WiFi 2.4GHz và Bluetooth được phép sử dụng tự do tại Việt Nam theo Thông tư 26/2016/TT-BTTTT với công suất phát tối đa 100mW. ESP32-CAM mặc định tuân thủ tiêu chuẩn này, không cần xin giấy phép sử dụng tần số.

Bảo mật dữ liệu cá nhân:

* Theo Luật An ninh mạng 2018 và Nghị định 13/2023/NĐ-CP, hệ thống ghi hình và nhận diện người cần có biển báo "Khu vực có camera giám sát" và chỉ lưu trữ video trong thời gian cần thiết (khuyến nghị < 30 ngày). Giải pháp: không lưu video dài hạn, chỉ streaming real-time, không sử dụng nhận diện khuôn mặt, mã hóa dữ liệu khi truyền.

Quyền riêng tư và giám sát:

* Theo Luật Dân sự 2015, không được giám sát khu vực riêng tư và cần có sự đồng ý của chủ sở hữu. Giải pháp: chỉ triển khai tại khu vực được phép (kho hàng, sân trường, khu công nghiệp) và có văn bản cho phép từ ban quản lý.

An toàn lao động:

* Theo Nghị định 44/2016/NĐ-CP, robot di động phải đảm bảo an toàn cho người xung quanh. Giải pháp: giới hạn tốc độ tối đa 0.5-1 m/s, cảm biến tránh vật cản bắt buộc, còi báo khi di chuyển, và nút dừng khẩn cấp.

# F. Phân tích ràng buộc kỹ thuật và môi trường

## I. Môi trường hoạt động

Nhiệt độ:

* Nhiệt độ hoạt động tối ưu: 20-35°C (môi trường ngoài trời điều kiện bình thường). ESP32-CAM có thể quá nhiệt khi streaming video liên tục ở nhiệt độ > 40°C, cần sử dụng tản nhiệt nhôm và tự động giảm FPS khi phát hiện quá nóng.

Độ ẩm:

* Độ ẩm hoạt động: 30-70% RH. Các linh kiện điện tử không chống nước có thể bị oxy hóa hoặc ngắn mạch ở độ ẩm cao, cần đóng gói trong hộp nhựa/acrylic và sử dụng gói hút ẩm silica gel.

Nhiễu sóng:

* Khoảng cách WiFi/Bluetooth bị giới hạn: 10-15m cho WiFi, 5-10m cho Bluetooth trong môi trường có vật cản. Nhiễu điện từ từ động cơ DC và thiết bị 2.4GHz khác có thể ảnh hưởng chất lượng tín hiệu, cần sử dụng tụ lọc chống nhiễu và chọn kênh WiFi ít chen lấn.

Nguồn cấp:

* Thời gian hoạt động: 1-2 giờ với pin Li-Po 7.4V 2200mAh. Cần sử dụng module BMS để bảo vệ pin, giám sát điện áp liên tục và tự động dừng xe khi pin < 15% để tránh xả sâu gây hỏng pin.

Ánh sáng:

* Chỉ hoạt động ban ngày hoặc môi trường có đủ ánh sáng (> 50 lux). Camera OV2640 không có đèn hồng ngoại nên chất lượng hình ảnh giảm mạnh trong điều kiện thiếu sáng, có thể bổ sung đèn LED trắng nếu cần hoạt động trong môi trường tối.

## II. Ràng buộc pháp lý

Quy định tần số vô tuyến:

* Sử dụng tự do không cần giấy phép. WiFi 2.4GHz và Bluetooth được phép sử dụng tại Việt Nam theo Thông tư 26/2016/TT-BTTTT với công suất tối đa 100mW, ESP32-CAM mặc định tuân thủ.

An toàn lao động:

* Tốc độ tối đa 0.5-1 m/s và có cảm biến tránh vật cản bắt buộc. Theo Nghị định 44/2016/NĐ-CP, robot phải đảm bảo an toàn với cơ chế dừng khẩn cấp khi phát hiện người/vật < 20cm.

## III. Tài nguyên thiết bị

Bộ nhớ ESP32-CAM:

* RAM 520KB + 4MB PSRAM không đủ để chạy AI. Bộ nhớ phải chia sẻ cho WiFi, Bluetooth, camera buffer và chương trình, do đó AI phải xử lý trên laptop thay vì trên ESP32-CAM.

CPU ESP32-CAM:

* Dual-core 240MHz không đủ mạnh cho xử lý AI real-time. Core 0 dành riêng cho WiFi/Bluetooth, Core 1 xử lý application, cần sử dụng RTOS task để quản lý đa nhiệm và ưu tiên task quan trọng.

GPIO giới hạn:

* Chỉ còn ~9 GPIO khả dụng sau khi camera và flash chiếm dụng. Cần thiết kế mạch cẩn thận để phân bổ GPIO cho động cơ, cảm biến, buzzer, và dự phòng cho mở rộng sau này.

# G. CÁC LÝ THUYẾT VÀ CÔNG NGHỆ ÁP DỤNG

## I. Kiến trúc IoT

### 1. Mô hình ba tầng (Three-tier Architecture)

Tầng cảm nhận (Perception Layer):

* Bao gồm các phần tử phần cứng như chip ESP32, camera OV2640 tích hợp trong module ESP32-CAM và cảm biến siêu âm HC-SR04. Tầng này có nhiệm vụ thu thập dữ liệu hình ảnh và khoảng cách từ môi trường thực tế.

Tầng mạng (Network Layer):

* Dữ liệu từ tầng cảm nhận được truyền qua WiFi 802.11n (2.4GHz) bằng giao thức WebSocket. Giao thức này giúp tạo kênh kết nối hai chiều giữa ESP32-CAM và laptop, cho phép truyền dữ liệu video, thông tin cảm biến và lệnh điều khiển theo thời gian thực mà không cần gửi nhiều yêu cầu như HTTP.

Tầng ứng dụng (Application Layer):

* Tầng này bao gồm laptop xử lý AI, ứng dụng di động điều khiển thiết bị và dashboard hiển thị trạng thái, cảnh báo. Laptop đảm nhiệm việc xử lý dữ liệu video, nhận dạng người, gửi lệnh điều khiển, đồng thời hiển thị kết quả lên giao diện người dùng.

### 2. Điện toán biên (Edge Computing)

Xử lý phân tán: Các tác vụ điều khiển động cơ, đo khoảng cách, và tránh vật cản được xử lý trực tiếp trên ESP32-CAM. Điều này giúp giảm độ trễ, tăng độ tin cậy và không phụ thuộc hoàn toàn vào kết nối mạng.

Xử lý trên máy tính/đám mây: Các tác vụ nặng như nhận diện người bằng AI (YOLO) được xử lý trên laptop, nơi có khả năng tính toán mạnh mẽ hơn (CPU/GPU).

II. Trí tuệ nhân tạo và Thị giác máy tính

### 1. Nhận diện đối tượng bằng YOLO (You Only Look Once)

Mô hình CNN (Convolutional Neural Network) nhận diện đối tượng trong thời gian thực.

Sử dụng YOLOv5 hoặc YOLOv8 được huấn luyện sẵn trên bộ dữ liệu COCO (80 lớp, trong đó có lớp “person”).

Là single-stage detector, YOLO xử lý xác định khung và phân loại trong một lần dự đoán, đạt tốc độ 10–30 FPS.

NMS (Non-Maximum Suppression) loại bỏ các khung chồng lặp, chỉ giữ kết quả có độ tin cậy cao nhất.

### 2. Thị giác máy tính với OpenCV

Xử lý khung hình (frame): giải mã luồng video MJPEG, chuyển đổi không gian màu, cắt và thay đổi kích thước ảnh.

Vẽ khung, nhãn và cảnh báo lên video bằng các hàm đồ họa của OpenCV.

Đọc, ghi và hiển thị video theo thời gian thực.

### 3. Xử lý ảnh trên ESP32-CAM

Mã hóa JPEG: nén khung hình từ định dạng raw xuống JPEG giúp giảm kích thước dữ liệu truyền (từ ~300KB xuống còn 30–50KB mỗi khung).

Tự động điều chỉnh độ sáng, độ tương phản và bão hòa để thích ứng với ánh sáng môi trường.

## III. Hệ thống nhúng và Vi điều khiển

ESP32 Microcontroller:

* Bộ xử lý Dual-core Xtensa LX6 32-bit @ 240MHz, một lõi xử lý WiFi/Bluetooth, lõi còn lại xử lý ứng dụng.
* Hệ điều hành FreeRTOS quản lý đa nhiệm (task scheduling, semaphore, queue).
* DMA (Direct Memory Access) giúp truyền dữ liệu từ camera trực tiếp vào bộ nhớ, giảm tải cho CPU.
* PWM (Pulse Width Modulation):
* Điều khiển tốc độ động cơ DC bằng cách thay đổi chu kỳ xung (duty cycle) từ 0–100%.
* Tần số PWM từ 1–20 kHz; duty cycle càng cao thì động cơ quay càng nhanh.
* ESP32 sử dụng bộ điều khiển LEDC (LED Controller) để tạo tín hiệu PWM.

Quản lý đa nhiệm (RTOS Task Management):

* Chạy song song nhiều tác vụ như camera, WiFi, WebSocket, động cơ, cảm biến.
* Ưu tiên các tác vụ quan trọng (điều khiển động cơ, dừng khẩn cấp).
* Giao tiếp giữa các tác vụ thông qua queue hoặc semaphore an toàn.

## IV. Giao thức truyền thông

### 1. Kết nối WiFi 802.11n (2.4GHz):

Áp dụng cơ chế CSMA/CA để tránh xung đột gói tin.

Sử dụng TCP/IP stack đảm bảo gói tin truyền đầy đủ, đúng thứ tự.

### 2. Giao thức WebSocket:

Handshake: ESP32-CAM khởi tạo kết nối bằng HTTP handshake để nâng cấp lên WebSocket.

Truyền dữ liệu thời gian thực: Cho phép gửi và nhận dữ liệu song công trên cùng một kênh duy nhất.

Định dạng dữ liệu: JSON, ví dụ:  
{"distance": 42.5, "alert": "on", "person\_count": 1}

Ưu điểm:

* Độ trễ thấp (5–20 ms).
* Tiết kiệm băng thông do không phải gửi lại header HTTP.
* Kết nối hai chiều, cho phép ESP32-CAM gửi video và cảm biến, đồng thời nhận lệnh điều khiển từ laptop.

### 3. Bluetooth Classic (BR/EDR):

Sử dụng SPP (Serial Port Profile) mô phỏng cổng nối tiếp để truyền lệnh ASCII.

Kiến trúc Master-Slave: điện thoại là master, ESP32-CAM là slave.

Hỗ trợ ghép nối và lưu bảo mật (pairing & bonding) bằng mã PIN.

### 4. Định dạng dữ liệu JSON:

Trao đổi dữ liệu giữa laptop và ESP32-CAM dạng:  
{"alert": "on", "person\_count": 2}

Dễ đọc, dễ phân tích, tương thích với Python và C/C++.

## V. Cảm biến và Điều khiển

Cảm biến siêu âm HC-SR04:

* Phát xung siêu âm 40kHz, đo thời gian phản hồi để tính khoảng cách:  
  Khoảng cách = (Thời gian × tốc độ âm thanh) / 2
* Độ chính xác ±3mm, phạm vi 2–400cm.

Bộ điều khiển động cơ L298N/TB6612FNG:

* Mạch cầu H (H-Bridge) giúp đảo chiều quay động cơ bằng cách thay đổi cực tính điện áp.
* Gồm 2 chân điều khiển hướng (IN1, IN2) và 1 chân PWM điều chỉnh tốc độ (ENA).

Nguyên lý:

* IN1 = HIGH, IN2 = LOW → quay thuận.
* IN1 = LOW, IN2 = HIGH → quay ngược.

Camera OV2640:

* Cảm biến CMOS 2MP (1600×1200), hỗ trợ nhiều độ phân giải.
* Giao tiếp qua cổng DVP 8-bit song song.
* Tự động cân bằng sáng và trắng (auto-exposure, auto-white balance).

## VI. Thuật toán và Xử lý

### 1. Thuật toán tránh vật cản:

Luật điều khiển:

* Nếu khoảng cách > 50cm → tiến bình thường (80% PWM).
* Nếu 20–50cm → giảm tốc (40% PWM).
* Nếu < 20cm → dừng, quay phải 90°, kiểm tra, nếu an toàn thì tiếp tục đi.
* Mô hình trạng thái:  
  Idle → Moving → Detecting Obstacle → Avoiding → Moving.

### 2. Bộ lọc Kalman:

Dùng để loại bỏ nhiễu cảm biến siêu âm bằng cách tính trung bình động của 5–10 mẫu gần nhất.

Ví dụ:  
filtered\_distance = (d1 + d2 + d3 + d4 + d5) / 5

## VII. Cập nhật Firmware OTA

Bộ nhớ flash chia làm hai vùng ứng dụng (mỗi vùng ~1.8MB).

Khi có bản cập nhật, ESP32-CAM tải firmware mới qua mạng, ghi vào vùng chưa kích hoạt.

Sau khi xác minh mã băm (MD5/SHA256) và khởi động lại, thiết bị chạy firmware mới.

Nếu lỗi trong ba lần khởi động, hệ thống tự động quay lại phiên bản cũ (rollback).

## VIII. Bảo mật

Mã hóa WPA2/WPA3: Sử dụng AES-128 hoặc AES-256 để bảo mật kết nối WiFi.

Kiểm tra toàn vẹn (Checksum Validation):

* So sánh mã băm MD5/SHA256 giữa firmware gốc và bản nhận.
* Nếu không trùng khớp, hệ thống từ chối cập nhật.

## IX. Công nghệ phát triển ứng dụng

### 1. Python với OpenCV và YOLO:

OpenCV: xử lý hình ảnh và video.

YOLOv8: nhận dạng người trong video real-time.

PyTorch/ONNX: nền tảng chạy mô hình học sâu.

### 2. Lập trình ESP32 với Arduino Framework:

Lập trình bằng C/C++ với API Arduino.

Dùng thư viện arduinoWebSockets hoặc ESPAsyncWebServer + AsyncWebSocket để kết nối WebSocket real-time.

Kết hợp FreeRTOS task để chia nhỏ luồng xử lý (camera, WebSocket, động cơ, cảm biến).

### 3. Ứng dụng di động:

Flutter (Dart): xây dựng app cho Android và iOS.

Sử dụng plugin web\_socket\_channel để kết nối trực tiếp tới ESP32 qua WebSocket.

### 4. Thư viện giao tiếp:

Python: sử dụng thư viện websockets để kết nối client tới ESP32.

ESP32: sử dụng AsyncWebSocket để tạo server WebSocket.

Ưu điểm: Truyền dữ liệu nhanh, hai chiều, hỗ trợ định dạng JSON, text hoặc nhị phân.