

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Bản đề xuất
**AutoEye: Hệ thống IoT Hỗ Trợ Đếm Và
Giám Sát Lưu Lượng Xe Lưu Thông
Trên Đường**

MÔN HỌC: Nhập môn lập trình điều khiển thiết bị thông minh

Sinh viên thực hiện:

Ngũ Kiệt Hùng (22127134)
Phạm Nguyên Hải Long (22127248)
Võ Thành Nghĩa (22127295)

Giảng viên:

ThS. Đỗ Thị Thanh Hà
TS. Võ Hoài Việt

Mục lục

1	Thông tin đồ án	3
1.1	Thông tin thành viên	3
1.2	Giới thiệu đồ án	3
2	Định hướng	4
2.1	Đặt vấn đề	4
2.2	Vị trí của sản phẩm	4
3	Mục Đích của sản phẩm	5
4	Mục tiêu của sản phẩm	5
5	Mô tả người dùng và các bên liên quan	7
5.1	Tổng quan về các bên liên quan	7
5.2	Tổng quan về người dùng	7
5.3	Môi trường sử dụng sản phẩm	7
5.4	Tổng quan các nhu cầu chính của bên liên quan hoặc người dùng	8
5.5	Các lựa chọn thay thế và đối thủ cạnh tranh	9
6	Tổng quan sản phẩm	9
6.1	Bối cảnh sản phẩm	9
6.2	Các giả định và sự phụ thuộc	9
6.3	Kiến trúc hệ thống	10
6.3.1	Sơ đồ kiến trúc hệ thống	10
6.3.2	Sơ đồ các đơn vị edge	11
6.4	Use case Diagram	12
6.4.1	Đặc tả cho từng use case	12
6.5	Quy trình tự động hoá tích hợp, cập nhật mô hình vào thiết bị	14
6.6	Thông tin thiết bị	15
7	Các tính năng của sản phẩm	15
8	Yêu cầu chức năng	16
8.1	Các đơn vị edge	16
8.2	Máy chủ MQTT	16
8.3	Máy chủ xử lý tương tác hệ thống - người dùng	16
8.4	Cơ sở dữ liệu	16
9	Các yêu cầu phi chức năng	17
10	Giải pháp đề xuất	18
10.1	Module rút trích đặc trưng	18
10.2	Module xác định vị trí	18
10.3	Module dự đoán trạng thái giao thông	18

10.4 Phương pháp đánh giá mô hình	19
11 Kế hoạch	19
11.1 Giai đoạn 1	19
11.2 Giai đoạn 2	19
11.3 Giai đoạn 3	20
11.4 Giai đoạn 4	20
12 Nguồn tham khảo	20

1 Thông tin đề án

1.1 Thông tin thành viên

Tên	MSSV	Email
Ngũ Kiệt Hùng	22127134	nkhung22@clc.fitus.edu.vn
Võ Thành Nghĩa	22127295	vtnghia22@clc.fitus.edu.vn
Phạm Nguyên Hải Long	22127248	pnhlong22@clc.fitus.edu.vn

1.2 Giới thiệu đề án

Mục đích của Vision Document là để thu thập, phân tích và xác định các yêu cầu cấp cao cùng những tính năng mong muốn cho một hệ thống giám sát giao thông sử dụng Edge AI. Hệ thống được thiết kế nhằm xây dựng một giải pháp giám sát tình trạng ùn tắc giao thông mà không sử dụng các dịch vụ của bên thứ 3 thông qua việc triển khai giải pháp phát hiện, phân loại và đếm số lượng phương tiện di chuyển qua một tuyến đường chính theo thời gian thực bằng trên các phần cứng nhỏ với chi phí thấp .

Tài liệu này ghi nhận các quan điểm và nhu cầu của các bên liên quan như quản lý giao thông đô thị và các nhà phát triển, nhằm cung cấp một cái nhìn rõ ràng về mục tiêu hệ thống và tính ứng dụng thực tế của nó trong các thành phố thông minh.

Tài liệu sẽ bắt đầu với phần Định hướng, được trình bày tại Mục 2. Định hướng, nơi nhóm phát triển sẽ hình thành vấn đề dựa trên yêu cầu của người dùng và các bên liên quan dưới dạng các tác vụ cụ thể. Tiếp theo, tài liệu sẽ cung cấp cái nhìn tổng quan về các bên liên quan và người dùng trong Mục 3. Mô tả người dùng và các bên liên quan, bao gồm các yêu cầu và ràng buộc từ phía họ.

Mục 4. Tổng quan sản phẩm sẽ trình bày khái quát về ứng dụng dịch vụ và mối quan hệ với các dịch vụ bên thứ ba, cũng như các giả định về sự phụ thuộc trong kiến trúc hệ thống. Mục 5. Các tính năng của sản phẩm sẽ mô tả chi tiết hơn về các tính năng cốt lõi được kỳ vọng trong phiên bản triển khai thực tế của dịch vụ. Cuối cùng, Mục 6. Yêu cầu chức năng sẽ xác định những yêu cầu về mặt vận hành của dịch vụ, các ràng buộc về thời gian hoạt động (uptime), cũng như hạ tầng vận hành được khuyến nghị cho người dùng và các yêu cầu dành cho người dùng cuối.

2 Định hướng

2.1 Đặt vấn đề

Mục	Nội dung
Vấn đề	Thiếu các giải pháp giám sát giao thông theo thời gian thực, tự động và có chi phí phải chăng dành cho các đoạn đường có lưu lượng cao.
Tác động lên	Tối ưu hóa chu kỳ đèn giao thông và ngăn ngừa tình trạng ùn tắc; hỗ trợ quy hoạch hạ tầng giao thông dài hạn.
Ảnh hưởng của vấn đề	Hạn chế việc sử dụng luồng dữ liệu giao thông liên tục; giảm tải các tác vụ tính toán không cần thiết cho máy chủ trung tâm.
Giải pháp sẽ là	Cung cấp một hạ tầng edge có chi phí thấp, đáng tin cậy, dễ mở rộng và dễ bảo trì, có khả năng cung cấp các cập nhật giao thông quan trọng chỉ khi cần thiết.

Bảng 1: Tổng quan vấn đề và giải pháp trong giám sát giao thông

2.2 Vị trí của sản phẩm

Mục	Nội dung
Dành cho	Chính quyền thành phố, cơ quan quản lý giao thông và các nhà phát triển hạ tầng thông minh.
Ai	Cần một hệ thống phân loại trạng thái giao thông và đếm phương tiện theo thời gian thực, có khả năng mở rộng và chi phí hợp lý.
Tên sản phẩm	AutoEye : Hệ thống IoT Hỗ Trợ Đếm Và Giám Sát Lưu Lượng Xe Lưu Thông Trên Đường.
Sử dụng	Sử dụng các mô hình AI (triển khai trên thiết bị biên hoặc máy chủ) để phát hiện và đếm phương tiện với độ trễ tối thiểu.
Không giống	Hệ thống truyền thống: Dựa trên camera giám sát (CCTV) kết hợp xử lý tập trung, đòi hỏi hạ tầng tốn kém.
Sản phẩm của chúng tôi	AutoEye sử dụng vi điều khiển chi phí thấp và suy luận AI tại thiết bị edge computing để xử lý video giao thông tại nguồn, giúp giảm băng thông và chi phí.

Bảng 2: Giới thiệu sản phẩm AutoEye và sự khác biệt

3 Mục Đích của sản phẩm

- **Giám sát giao thông:** Cho phép báo cáo tình trạng lưu thông của phương tiện theo thời gian thực, phục vụ cho việc điều tiết giao thông (cả tự động và thủ công) và quản lý ùn tắc mà không cần phụ thuộc vào dịch vụ của các bên thứ ba.
- **Phát triển thành phố thông minh:** Thực hiện số hóa và thông minh hóa các hạ tầng thị giác hiện tại của thành phố (camera). Dự án đồng thời xây dựng một “bản thảo tương tác” (interface) nhằm giúp các dịch vụ nội bộ của hệ thống quản lý thành phố kết nối hiệu quả hơn với hạ tầng của các điểm gắn camera.
- **Cung cấp dữ liệu giao thông:** Cung cấp dữ liệu giao thông ở cấp độ thứ cấp để phục vụ cho các tác vụ yêu cầu dữ liệu lớn trong tương lai, đặc biệt trong bối cảnh phát triển mạnh mẽ của các phương pháp học máy yêu cầu dữ liệu lớn.

Sau cùng, hệ thống nhắm đến một mục đích lớn hơn là từng bước hiện đại hóa, thông minh hóa các hạ tầng giao thông của đô thị; trong đó, sự “thông minh” chỉ đến khả năng thích ứng linh động của hạ tầng về mặt cài đặt (như thời gian đèn tín hiệu giao thông, thông báo phân loại ùn tắc giao thông, ...) nhằm giúp ích, giảm thiểu

4 Mục tiêu của sản phẩm

Mục Tiêu 1: Xây dựng hệ thống nhận diện và đếm phương tiện có thể hoạt động ổn định trong điều kiện thực tế

Các tiêu chí	Mô tả
Specific	Mục tiêu ở đây là sẽ xây dựng được hệ thống nhận diện và đếm phương tiện.
Measurable	Độ chính xác mong muốn trên 90%, nhận diện được các loại xe và đếm các loại phương tiện một cách chính xác.
Achievable	Mục tiêu có thể đạt được nếu nhóm sử dụng công nghệ phù hợp và có đủ thời gian & tài nguyên.
Relevant	Với các thiết bị chi phí thấp và những nghiên cứu liên quan hỗ trợ, đồ án có tính khả thi (thực hiện được).
Time Bound	Đã đặt deadline cho mục tiêu trong timeline.

Bảng 3: Bảng tiêu chí SMART cho dự án

Mục Tiêu 2: Thiết kế và xây dựng được hệ thống có tích hợp cả cơ sở dữ liệu, giao diện người dùng để bên phía người dùng cuối có thể sử dụng để quan sát được tình trạng giao thông

Các tiêu chí	Mô tả
Specific	Phát triển phần mềm trên máy chủ để hỗ trợ tương tác với người dùng và cơ sở dữ liệu.
Measurable	Đảm bảo cập nhật tình trạng giao thông thời gian thực, không quá 30 giây so với tình trạng giao thông ngoài thực tế.
Achievable	Kế hoạch được hoạch định và chia nhỏ cho các thành viên.
Relevant	Quy mô của kiến trúc hệ thống vừa phải, sử dụng công nghệ mở, nên mục tiêu khả thi.
Time Bound	Đã đặt deadline cho mục tiêu trong timeline.

Bảng 4: Bảng tiêu chí SMART cho phần mềm máy chủ

Mục tiêu 3: Có thể triển khai mô hình AI trên thiết bị phần cứng với dung lượng và tài nguyên hạn chế, cụ thể là ESP32-CAM.

Các tiêu chí	Mô tả
Specific	Triển khai giải pháp phần mềm trên các phần cứng IoT.
Measurable	<ul style="list-style-type: none"> Độ chính xác = 70%–80% (vì triển khai trên phần cứng IoT) Thành công = 85% số khung hình trong điều kiện ánh sáng ban ngày Triển khai được mô hình trên thiết bị có bộ nhớ flash < 16MB
Achievable	Kế hoạch cần chi tiết và phù hợp với các thành viên trong nhóm.
Relevant	Phần cứng không quá phức tạp, có sự hỗ trợ của các thư viện nạp mã nguồn nên mục tiêu khả thi.
Time Bound	Đã đặt deadline cho mục tiêu trong timeline.

Bảng 5: Bảng tiêu chí SMART cho phần mềm trên thiết bị IoT

Tên	Mô tả	Trách nhiệm
Người tham gia giao thông	Sử dụng hệ thống qua giao diện người dùng	Cung cấp nhận xét về những tính năng bị lỗi, hoặc những tính năng chưa đáp ứng được yêu cầu.
Ph.D Võ Hoài Việt	Giám sát dự án	Đánh giá và nhận xét về những yêu cầu của dự án. Giám sát và hỗ trợ trong quá trình phát triển dự án và đảm bảo hoàn thành tiến độ.
Đội ngũ phát triển	Thiết kế kiến trúc hệ thống, thiết kế giải pháp phần mềm dựa vào những yêu cầu sẵn có	Xây dựng, vận hành và giám sát hệ thống IoT và hệ thống phần mềm.

Bảng 6: Bảng mô tả vai trò và trách nhiệm của các bên liên quan

5 Mô tả người dùng và các bên liên quan

5.1 Tổng quan về các bên liên quan

5.2 Tổng quan về người dùng

Tên	Mô tả	Trách nhiệm
Người tham gia giao thông	Sử dụng hệ thống qua giao diện người dùng	Cung cấp nhận xét về những tính năng bị lỗi, hoặc những tính năng chưa đáp ứng được yêu cầu.
Ph.D Võ Hoài Việt	Giám sát dự án	Đánh giá và nhận xét về những yêu cầu của dự án. Giám sát và hỗ trợ trong quá trình phát triển dự án và đảm bảo hoàn thành tiến độ.
Đội ngũ phát triển	Thiết kế kiến trúc hệ thống, thiết kế giải pháp phần mềm dựa vào những yêu cầu sẵn có	Xây dựng, vận hành và giám sát hệ thống IoT và hệ thống phần mềm.

Bảng 7: Bảng mô tả vai trò và trách nhiệm của các bên liên quan

5.3 Môi trường sử dụng sản phẩm

Thiết bị và Truy cập:

- **Thiết bị sử dụng:**
 - Module ESP32-CAM được lắp đặt tại hiện trường để ghi lại hình ảnh giao thông và thực hiện các tác vụ theo dõi, phân loại phương tiện.
 - Máy chủ cục bộ hoặc đám mây dùng để xử lý và lưu trữ dữ liệu.

- Giao diện bảng điều khiển (Dashboard): có thể truy cập thông qua trình duyệt trên máy tính xách tay, máy tính bảng hoặc máy tính để bàn.
- **Kết nối Internet:**
 - Các thiết bị kết nối qua Wifi để truyền hình ảnh hoặc kết quả suy luận.
 - Hệ thống cần được tối ưu hóa để hoạt động hiệu quả trong điều kiện mạng chậm chèn hoặc băng thông thấp.

Nền tảng hệ thống:

- **Các nền tảng hiện tại:**
 - Firmware được phát triển bằng TinyEngine phiên bản 2.0 cho ESP32-CAM.
 - Máy chủ sử dụng giao thức MQTT để giao tiếp.
 - Cơ sở dữ liệu sử dụng MongoDB.
 - Máy chủ tương tác người dùng - hệ thống sử dụng React.js cho giao diện người dùng và FastAPI cho các tác vụ xử lý yêu cầu người dùng.

5.4 Tổng quan các nhu cầu chính của bên liên quan hoặc người dùng

Nhu cầu	Ưu tiên	Vấn đề	Giải pháp hiện tại	Giải pháp đề xuất
Đếm phương tiện theo thời gian thực	High	Thiết bị nhỏ (Tiny device) có khả năng tính toán hạn chế.	Tính toán thông qua điện toán đám mây (over-the-cloud).	Theo dõi phương tiện hiệu quả bằng CNN trên các thiết bị edge.
Phân loại tình trạng giao thông	High	Thiết bị nhỏ có khả năng xử lý hạn chế.	Kết nối tới các dịch vụ trung tâm như Google Maps để đếm lưu lượng tại các điểm.	Kiến trúc phân loại hiệu quả sử dụng các đặc trưng trích xuất từ CNN.
Giao diện người dùng	Medium	Kết nối Internet có thể không ổn định. Khả năng tích hợp vào thiết bị edge có thể không khả thi.	Mô-đun mạng chung được sử dụng bởi tất cả các thiết bị hỗ trợ Arduino.	Mô-đun mạng nhẹ do nhà cung cấp ESP phát triển.
Khả năng mở rộng hệ thống	Medium	Máy chủ tính toán trung tâm trở thành điểm nghẽn.	Hệ thống phụ thuộc vào tính toán đám mây tập trung.	Thực hiện tính toán trực tiếp trên thiết bị (on-device computation).

Bảng 8: Phân tích nhu cầu và giải pháp cho hệ thống giám sát giao thông

5.5 Các lựa chọn thay thế và đối thủ cạnh tranh

Hệ Thống	Điểm mạnh	Điểm yếu
Camera giám sát truyền thống + AI trên đám mây	Độ chính xác cao	Chi phí cao, yêu cầu băng thông lớn
Đếm thủ công	Chi phí thấp	Tốn nhân lực, không thể mở rộng quy mô
Các nhà cung cấp đèn giao thông thông minh (ví dụ: Hikvision AI Cam)	Hệ thống đạt chuẩn thương mại	Rất đắt đỏ, phụ thuộc vào nhà cung cấp (vendor lock-in)

Bảng 9: So sánh các giải pháp giám sát giao thông hiện tại

6 Tổng quan sản phẩm

6.1 Bối cảnh sản phẩm

Phần này cung cấp cái nhìn tổng quan về các khả năng của sản phẩm, giao tiếp với các ứng dụng khác và cấu hình hệ thống. AutoEye là đặc tả và hiện thực hóa một hệ thống đề xuất nhằm theo dõi phương tiện và phân loại trạng thái giao thông ngay trên thiết bị, áp dụng cho một đoạn đường cụ thể. Hệ thống hướng tới việc thiết kế một giải pháp chi phí thấp, hiệu suất cao, triển khai các thuật toán học máy ngay trên thiết bị, nhằm hỗ trợ khả năng mở rộng theo chiều ngang khi áp dụng vào hạ tầng ở cấp độ thành phố.

6.2 Các giả định và sự phụ thuộc

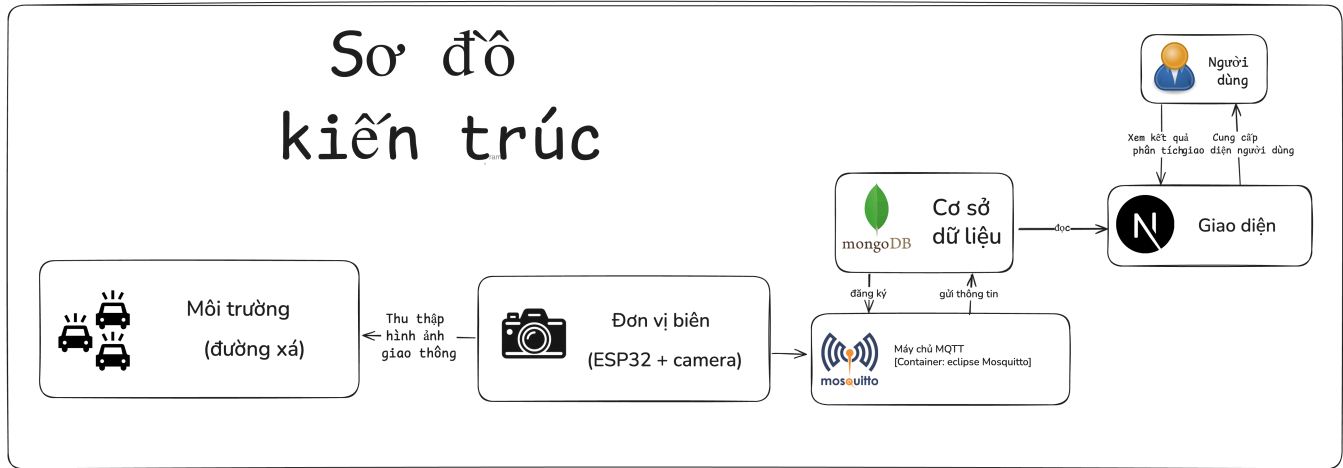
Những sự giả định và phụ thuộc của AutoEye bao gồm:

- **Môi trường vận hành:** Giả định rằng AutoEye sẽ được triển khai trên các vi xử lý có tài nguyên hạn chế nghiêm trọng, với dung lượng bộ nhớ dưới 512 KiB và bộ nhớ flash chỉ cho phép đọc. Hệ thống cũng bao gồm một phía máy chủ, có chức năng giám sát và tổng hợp các kết quả được báo cáo từ các vi xử lý đã triển khai, nhằm cung cấp góc nhìn phân tích dữ liệu.
- **Phụ thuộc bên ngoài:** AutoEye được thiết kế nhằm xây dựng một hệ thống lưu trữ và truyền thông dữ liệu hoàn chỉnh, không phụ thuộc vào các thành phần hoặc dịch vụ bên ngoài.
- **Khả năng mở rộng:** Hệ thống được giả định có thể xử lý hiệu quả một lượng nhỏ dữ liệu trong khoảng thời gian định sẵn. Mỗi thiết bị biên (edge unit) phải tự thực hiện các tác vụ theo dõi và phân loại mà không cần chuyển giao việc xử lý cho các thiết bị khác.

Nếu những yếu tố này thay đổi, có thể sẽ cần điều chỉnh hoặc cập nhật Tài liệu Tổng quan để đáp ứng các yêu cầu mới.

6.3 Kiến trúc hệ thống

6.3.1 Sơ đồ kiến trúc hệ thống



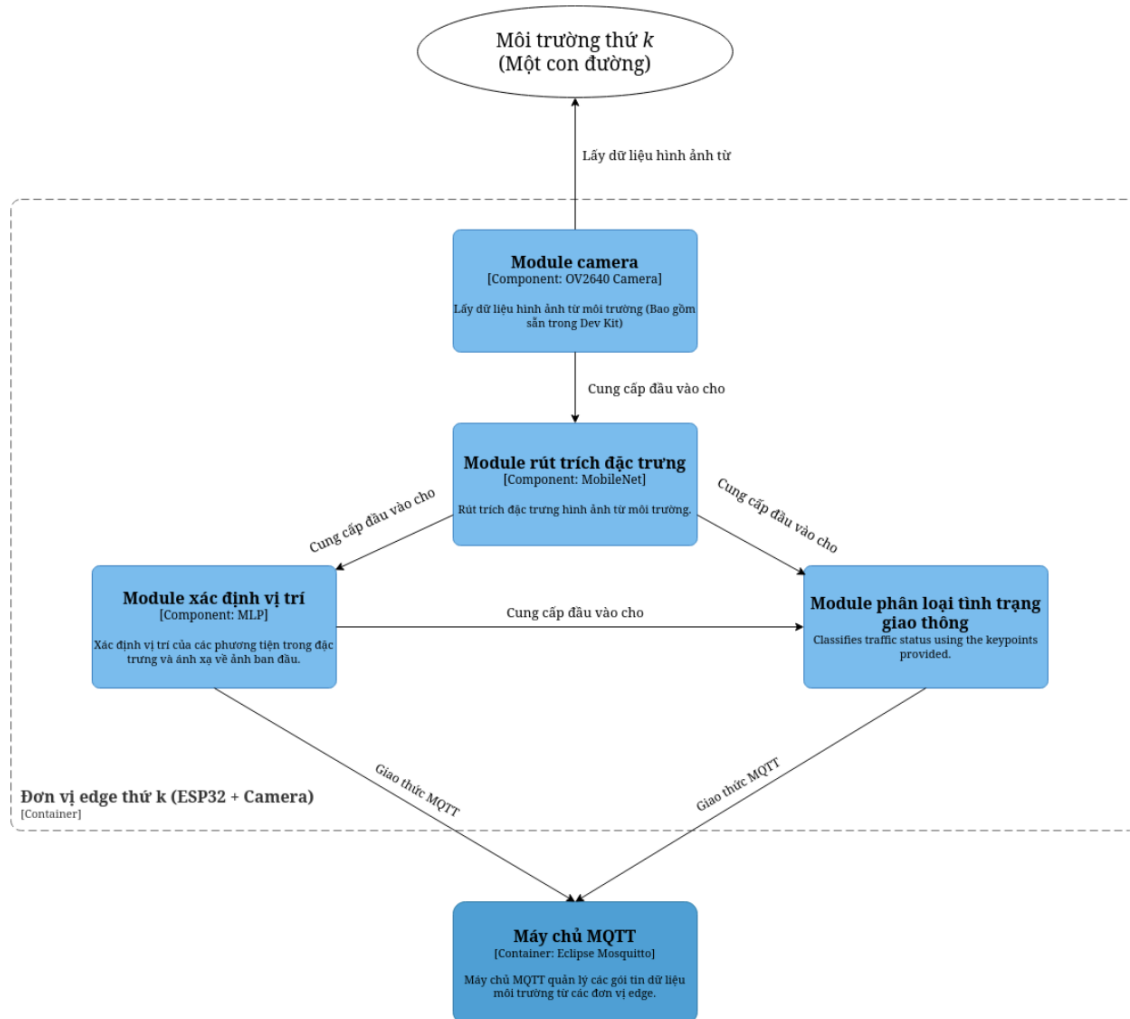
Hình 1: Sơ đồ hệ thống

Kiến trúc hệ thống chung của AutoEye bao gồm các đơn vị edge trực tiếp thu thập dữ liệu từ môi trường và phân tích dữ liệu để đưa ra các thông tin về lưu lượng xe, tình trạng ùn tắc giao thông tại các con đường.

Các đơn vị edge sẽ độc lập xử lý dữ liệu của khu vực mà nó được phân bổ để giải quyết. Các kết quả từ mỗi đơn vị edge sẽ được gửi đến một máy chủ trung tâm thông qua giao thức MQTT. Thông tin gửi đến của mỗi đơn vị edge sẽ bao gồm chỉ số định danh của đơn vị edge, thời gian gửi, lượng xe tại thời điểm đó và tình trạng ùn tắc giao thông. Các dữ liệu này sẽ được lưu trữ tạm thời trên máy chủ MQTT.

Sau mỗi khoảng thời gian T, một máy chủ khác sẽ yêu cầu lượng dữ liệu đang được chứa trong hàng chờ của máy chủ MQTT thông qua giao thức HTTP. Máy chủ này mang vai trò quản lý xuất-nhập của dữ liệu vừa thu được từ các đơn vị edge thông qua máy chủ MQTT với một cơ sở dữ liệu; đồng thời cung cấp và tương tác với giao diện người dùng khi cần thiết. Máy chủ cho phép xem tình trạng mới nhất của mỗi đơn vị edge, bao gồm cả các đơn vị có thể mất kết nối. Ngoài ra, máy chủ còn cho phép truy xuất các thông tin đã được lưu lại từ trước.

6.3.2 Sơ đồ các đơn vị edge



Các đơn vị edge có vai trò thu thập các dữ liệu của môi trường đang hoạt động (là các con đường) thông qua module camera OV2640.

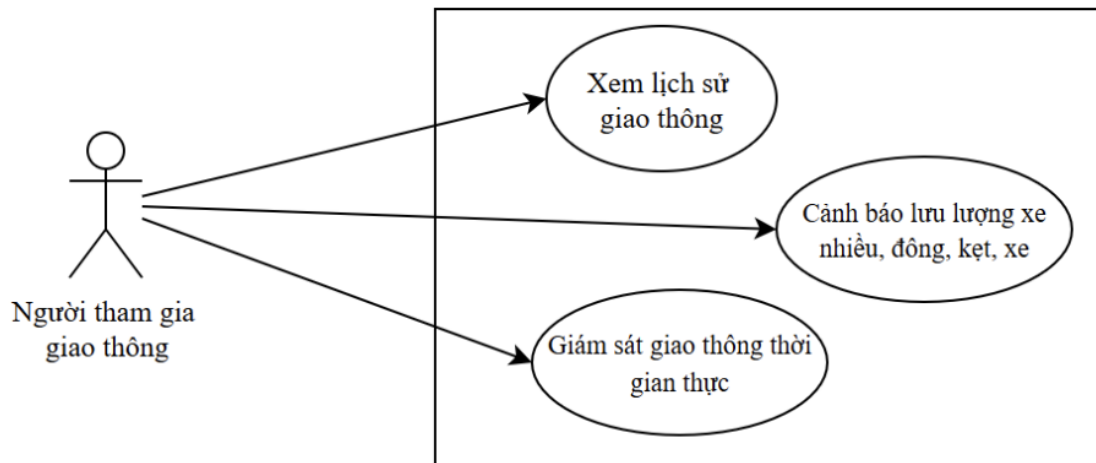
Sau đó, các đơn vị edge sẽ độc lập sử dụng mô hình MobileNet cùng với một số các kỹ thuật tối ưu bộ nhớ sử dụng để rút trích và giảm chiều các đặc trưng của các ảnh từ môi trường.

Tiếp theo, đặc trưng từ module rút trích đặc trưng sẽ được cung cấp cho module xác định vị trí (localizer) nhằm dự đoán vị trí của các phương tiện bằng một mạng MLP thông thường.

Đầu ra của module dự đoán vị trí và rút trích đặc trưng sẽ được cung cấp cho module dự đoán trạng thái giao thông. Module dự đoán trạng thái giao thông sẽ tiến hành phân loại tình trạng giao thông theo lưu lượng xe. Cụ thể, module sẽ xét số phương tiện đang lưu hành trong t khung hình gần nhất và tốc độ di chuyển trung vị của các phương tiện đó nhằm xác định tình trạng có ùn tắc giao thông, có nguy cơ ùn tắc giao thông hoặc không có nguy cơ ùn tắc giao thông.

6.4 Use case Diagram

Use Case Diagram Cho Hệ Thống Auto Eye



6.4.1 Đặc tả cho từng use case

1. Use case 1: Xem lịch sử giao thông

Thành phần	Mô tả
Tên	Xem lịch sử giao thông
Tác nhân	Người tham gia giao thông
Mục tiêu	Truy xuất dữ liệu lưu lượng phương tiện trong một khoảng thời gian
Cơ sở dữ liệu	Bảng <code>traffic_log</code>

Luồng xử lý:

- Người dùng chọn khu vực cần xem và thời gian cần thống kê (ví dụ: từ 6h đến 9h ngày 28/5).
- Hệ thống thực hiện truy vấn đến bảng `traffic_log`.
- Dữ liệu trả về dưới dạng bảng hoặc biểu đồ đường.
- Hiển thị xu hướng lưu lượng xe theo từng khung giờ, từng loại xe.

2. Use case 2: Cảnh báo lưu lượng xe nhiều, đông, kẹt xe

Thành phần	Mô tả
Tên	Cảnh báo lưu lượng xe nhiều, đông, kẹt xe
Tác nhân	Người tham gia giao thông
Mục tiêu	Nhận cảnh báo khi lưu lượng phương tiện vượt quá ngưỡng cho phép
Cơ sở dữ liệu	Bảng <code>traffic_alerts</code> , <code>thresholds</code> (mức ngưỡng)

Luồng xử lý:

- Sau khi AI model đếm phương tiện, kết quả được so sánh với mức ngưỡng (threshold) theo từng khung giờ/khu vực.
- Nếu số lượng xe trong khung giờ vượt quá ngưỡng, hệ thống tạo một bản ghi cảnh báo.
- Gửi thông báo (qua API/Telegram/email) đến người dùng hoặc hệ thống dashboard.
- Người tham gia giao thông thấy cảnh báo đỏ, khu vực kẹt xe → chủ động thay đổi tuyến đường.

3. Use case 3: Giám sát giao thông thời gian thực

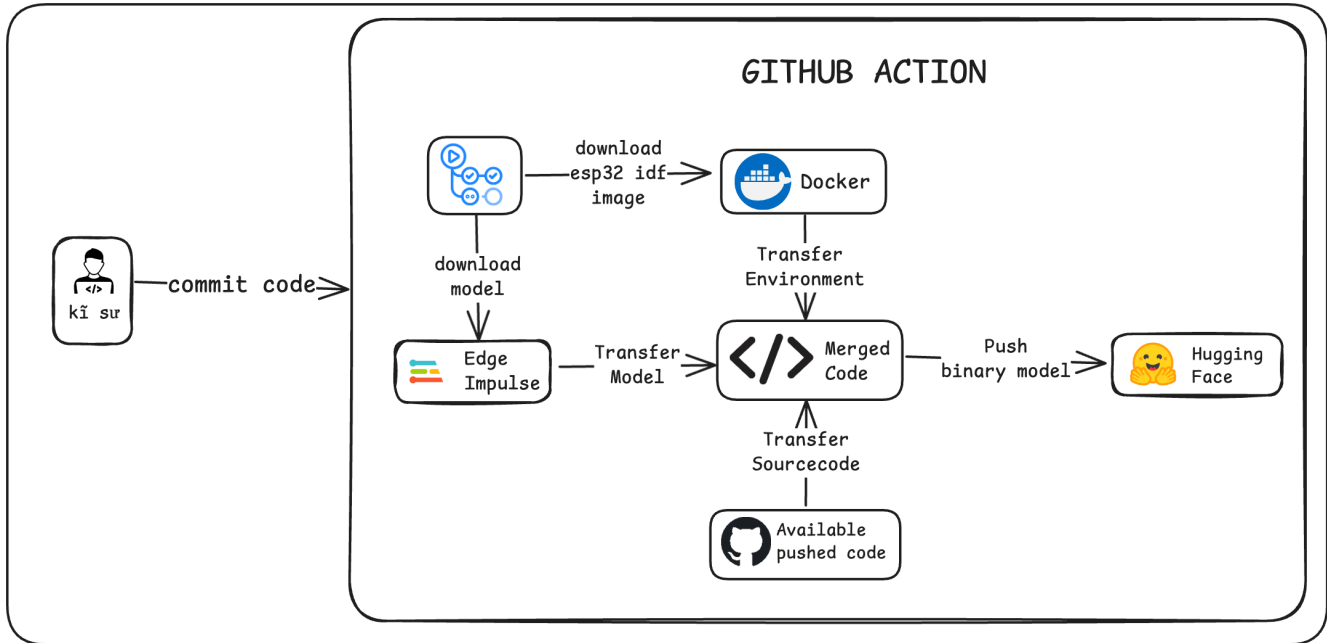
Bảng 10: Use case: Giám sát giao thông thời gian thực

Thành phần	Mô tả
Tên	Giám sát giao thông thời gian thực
Tác nhân	Người tham gia giao thông
Mục tiêu	Hiển thị số lượng xe đang lưu thông tại một khu vực theo thời gian thực
Cơ sở dữ liệu	Bảng <code>traffic_log</code> (ghi lại số lượng xe, thời gian, vị trí)

Luồng xử lý:

- Thiết bị ESP32-CAM chụp hình giao thông định kỳ (mỗi 5 giây).
- Ảnh được xử lý tại Edge:
 - Ảnh được tiền xử lý bóc tách đặc trưng
 - Model AI nhận diện phương tiện và đếm số lượng
- Kết quả được ghi vào DB (`traffic_log` với các trường `time_stamp`, `location_id`, `vehicle_count`, `vehicle_type`).
- Người dùng truy cập dashboard để truy xuất dữ liệu `traffic_log` gần nhất.
- Hiển thị biểu đồ lưu lượng giao thông trực tiếp trên dashboard.

6.5 Quy trình tự động hoá tích hợp, cập nhật mô hình vào thiết bị



Hình 2: Sơ đồ tự động hoá tích hợp và cập nhật mô hình vào thiết bị

Để tự động hoá quá trình tích hợp và cập nhật mô hình vào thiết bị, chúng tôi sẽ sử dụng một quy trình CI/CD (Continuous Integration/Continuous Deployment) với các bước như sau:

- Mỗi lần cập nhật mô hình tại Edge Impulse, kĩ sư chỉ cần commit viết vào readme của dự án trên GitHub (có thể là cải thiện mô hình, cập nhật mô hình mới, hoặc thay đổi cấu trúc dự án).
- GitHub Actions sẽ tự động kéo 3 thành phần bao gồm
 - Mô hình đã được huấn luyện từ Edge Impulse.
 - Môi trường biên dịch mô hình từ Dockerhub của ESP-IDF.
 - Cùng với các tập tin cấu hình cần thiết để biên dịch mô hình ở Github.
- Sau đó, GitHub Actions sẽ biên dịch mô hình và tạo ra một tập tin nhị phân (binary) có thể chạy trên thiết bị ESP32.
- Tập tin nhị phân này sẽ được đẩy lên Huggingface để lưu trữ phiên bản mới nhất của mô hình.

Từ đây ở phía ESP32, chúng tôi sẽ sử dụng một đoạn mã để tự động tải xuống mô hình mới nhất từ Huggingface và cập nhật mô hình trên thiết bị. Điều này cho phép chúng tôi dễ dàng cập nhật mô hình mà không cần phải can thiệp thủ công vào từng thiết bị.

6.6 Thông tin thiết bị

Bảng 11: Danh sách thiết bị và thông số với nguồn mua

Tên thiết bị	Giá tiền	Bộ nhớ chính (RAM)	Bộ nhớ phụ	Nguồn mua	Ghi chú
ESP32 CAM 16 MB Flash	255.000 VND	512 KB SRAM + 8 MB	16 MB	LILYGO® ESP32-Cam	Linh kiện dự phòng
Raspberry Pi Zero 2	610.000 VND	512 MB DDR2 RAM	—	Raspberry Pi Zero 2 W (Wireless)	Linh kiện dự phòng
MicroSD San-disk	140.000 VND	—	32 GB	Sandisk 32 GB	Linh kiện dự phòng

7 Các tính năng của sản phẩm

Bảng 12: Danh sách chức năng hệ thống

Chỉ mục	Tên	Mô tả	Độ ưu tiên
1	Đăng ký thiết bị	Đăng ký các thiết bị IoT mới (ví dụ: ESP32-CAM) và chỉ định các vùng quan sát.	Cao
2	Phát hiện các phương tiện (theo thời gian thực)	Phát hiện và đếm các phương tiện đi qua vùng quan sát camera ở thời gian thực.	Cao
3	Phân loại tình hình giao thông (theo thời gian thực)	Sử dụng thuật toán đếm phương tiện cơ bản hoặc mạng MLP động để phân loại tình hình giao thông tại vị trí quan sát.	Cao
4	Truyền dữ liệu	Truyền dữ liệu quan sát thông qua MQTT/HTTP tới máy chủ trung tâm hoặc nền tảng đám mây.	Cao
5	Bảng theo dõi dữ liệu	Trực quan hoá số lượng, xu hướng và phân tích giao thông thông qua giao diện website.	Cao
6	Truy vấn lịch sử dữ liệu	Truy cập các dữ liệu giao thông được lọc qua thời gian, địa điểm hoặc loại phương tiện.	Trung bình
7	Hệ thống báo động	Kích hoạt cảnh báo khi mật độ giao thông vượt quá ngưỡng đã được xác định trước.	Trung bình
8	Theo dõi tình trạng của thiết bị	Hiển thị khả năng kết nối, trạng thái và lỗi từ các thiết bị biên.	Trung bình

8 Yêu cầu chức năng

8.1 Các đơn vị edge

- **Có thể thu thập dữ liệu hình ảnh từ môi trường:** các thiết bị của mỗi đơn vị edge phải bao gồm bộ phận cho phép thu thập hình ảnh.
- **Có thể thực hiện đo lưu lượng xe từ môi trường:** các đơn vị edge phải báo cáo được lưu lượng xe trong một khoảng thời gian t đã xác định trước. Hiện tại, yêu cầu cho độ chính xác là trên 50%.
- **Có thể thực hiện phân loại tình trạng giao thông đúng:** các đơn vị edge phải báo cáo phân loại tình trạng giao thông, gồm 3 mức: {Ùn tắc, có nguy cơ ùn tắc, không có ùn tắc}.
- **Có thể báo cáo tình trạng sống/còn của thiết bị:** các đơn vị edge phải có cơ chế giao tiếp với máy chủ MQTT nhằm báo cáo tình trạng mất kết nối/còn kết nối của mỗi thiết bị.

8.2 Máy chủ MQTT

- **Có thể tiếp nhận và lưu trữ dữ liệu:** máy chủ MQTT phải đáp ứng yêu cầu tiếp nhận và lưu trữ các dữ liệu được gửi lên từ các đơn vị edge vào một hàng chờ chính mà không theo trình tự nào. Tính nhất quán của dữ liệu phải được bảo toàn.
- **Có thể tiếp nhận yêu cầu từ máy chủ khác và trả về dữ liệu tạm thời:** máy chủ MQTT phải sẵn sàng tiếp nhận các yêu cầu truy xuất dữ liệu từ máy chủ khác và trả về toàn bộ dữ liệu đang lưu trữ tạm thời từ hàng chờ chính.
- **Có thể được tìm thấy qua Internet:** máy chủ MQTT phải có thể tìm được thông qua Internet.

8.3 Máy chủ xử lý tương tác hệ thống - người dùng

- **Có thể kết nối và truy xuất dữ liệu đến máy chủ MQTT:** máy chủ xử lý tương tác người dùng - hệ thống phải đảm bảo có đường liên kết đến máy chủ MQTT.
- **Có thể kết nối, truy xuất và thêm dữ liệu đến cơ sở dữ liệu:** máy chủ xử lý tương tác người dùng - hệ thống phải đảm bảo có liên kết đến ít nhất một cơ sở dữ liệu và có thể truy xuất hoặc chỉnh sửa dữ liệu trong cơ sở dữ liệu.
- **Có thể tiếp nhận yêu cầu hiển thị thông tin mới nhất cho người dùng:** máy chủ xử lý tương tác người dùng - hệ thống phải đảm bảo có thể cung cấp các dữ liệu mới nhất về tất cả đơn vị edge khi người dùng yêu cầu. Đồng thời cung cấp giao diện để hiển thị các dữ liệu đó.
- **Có thể tiếp nhận yêu cầu hiển thị thông tin cũ:** máy chủ xử lý tương tác người dùng - hệ thống phải đảm bảo có thể cho phép người dùng truy xuất các dữ liệu đã được thu thập từ mỗi đơn vị edge trước đây.

8.4 Cơ sở dữ liệu

- **Có thể kết nối đến máy chủ xử lý tương tác hệ thống - người dùng:** cơ sở dữ liệu phải đảm bảo rằng có ít nhất một liên kết đến máy chủ xử lý tương tác hệ thống - người

dùng.

- **Đảm bảo các tác vụ ghi được tối ưu về mặt thời gian:** cơ sở dữ liệu phải đảm bảo về việc các tác vụ ghi sẽ là các tác vụ chính và cần tối ưu. Thời gian tối đa cho mỗi yêu cầu ghi tùy thuộc vào thiết kế hệ thống.

9 Các yêu cầu phi chức năng

1. Hiệu suất

- Hệ thống phải chạy và trả ra các kết quả phát hiện trong vòng 1–2 giây mỗi ảnh (ESP hoặc từ server).
- Hệ thống nên hỗ trợ ít nhất 20 thiết bị biên mỗi địa điểm mà không có độ trễ.

2. Khả năng sử dụng

- Hệ thống phải có khả năng mở rộng để hỗ trợ hàng chục đến hàng trăm nút giao thông với cấu hình nhỏ nhất.
- Hệ thống cần hỗ trợ khả năng mở rộng theo chiều ngang trong quá trình thu thập và trực quan hóa dữ liệu, cho phép dễ dàng bổ sung thêm thiết bị hoặc nút mới mà không ảnh hưởng đến hiệu suất tổng thể.

3. Bảo mật

- Giao tiếp giữa các thiết bị và servers phải được bảo mật thông qua TLS 1.2 hoặc cao hơn.
- Các thiết bị nên sử dụng khóa chia sẻ trước hoặc chứng chỉ để ngăn chặn việc tiêm dữ liệu trái phép.

4. Độ tin cậy

- Hệ thống nên khôi phục sau khi thiết bị bị gián đoạn bằng cách sử dụng cơ chế thử lại hoặc kết nối lại.
- Mục tiêu duy trì thời gian hoạt động 99,5% đối với các dịch vụ cốt lõi.

5. Khả năng bảo trì

- Mã nguồn phải được thiết kế theo mô-đun với các thành phần riêng biệt cho:
 - AI biên (Edge AI).
 - Dòng xử lý dữ liệu (Data pipeline).
 - Bảng dữ liệu / Frontend.
- Phải cung cấp tài liệu rõ ràng cho các nhà phát triển (developers) và người vận hành (operators).

6. Khả năng tương tác

- Hệ thống nên hỗ trợ giao tiếp thông qua MQTT, HTTP hoặc WebSocket.
- Dễ dàng tích hợp với các hệ thống thành phố khác (ví dụ: Smart City platform, bộ điều khiển đèn giao thông) thông qua API.

10 Giải pháp đề xuất

10.1 Module rút trích đặc trưng

Nhằm đáp ứng được nhu cầu về phần cứng có giới hạn của các đơn vị edge, nhóm đề xuất sử dụng mô hình MobileNet [5] nhờ kiến trúc tách rời các lớp tích chập thành hai bước nhỏ hơn, từ đó giảm mức dung lượng tối đa (peak memory) mà mô hình cần.

Tiếp theo, nhóm thực hiện tối ưu mô hình dựa trên các kỹ thuật cân bằng dung lượng sử dụng tại mỗi lớp mô hình, được trình bày trong MCUNetV2 [2]. Đồng thời, một kỹ thuật tìm kiếm kiến trúc tối ưu TinyNAS [2] được trình bày trong MCUNetV2 cũng được nhóm tìm hiểu nhằm tiếp tục tối ưu độ chính xác của mô hình.

Nhóm dự kiến sẽ sử dụng tập dữ liệu Stanford Cars và CompCars [3]. Nhóm sẽ sử dụng các kỹ thuật lượng hóa các tham số mô hình về 8-bit [4] cho cả trọng số và đầu ra ở mỗi lớp của mô hình. Mô hình sẽ được huấn luyện trên TensorFlow, sau đó được chuyển qua TensorFlow Lite và biên dịch theo chương trình có sẵn của TinyEngine.

10.2 Module xác định vị trí

Để thực hiện xác định vị trí của các phương tiện, nhóm sẽ sử dụng lớp dự đoán tương tự như cấu trúc đã trình bày.

Tất nhiên, vì mô hình được trình bày trong bài báo đã được huấn luyện trên một tập dữ liệu có rất nhiều lớp phải đoán, cùng kiến trúc có hạn, kết quả không quá tốt. Tuy nhiên, nhóm sẽ ứng dụng kiến trúc cho một tập dữ liệu chủ đạo về phương tiện di chuyển, cũng như cho phép mô hình học đoán không cần quá chính xác bounding box của các phương tiện, mà chỉ cần vị trí tương đối và độ tin cậy cao cho dự đoán.

10.3 Module dự đoán trạng thái giao thông

Để thực hiện dự đoán có tính tiên định, nhóm sẽ dùng phương pháp đơn giản là chia khung hình thu được thành 2 phần bằng nhau theo chiều dọc. Sau đó, với t khung hình gần nhất, các đơn vị edge sẽ thực hiện gom hai khung hình liền kề (tức gom khung hình t với khung hình $t-1$, khung hình $t-1$ với khung hình $t-2$, ...) và thực hiện các phép toán sau với mỗi cặp khung hình:

Với i là cặp thứ i trong các cặp đã hình thành, ta lấy số lượng phương tiện bên nửa trái của khung hình đầu tiên $s11[i]$, số lượng phương tiện nằm bên nửa phải khung hình đầu tiên: $s12[i]$. Ta tiếp tục lấy số lượng phương tiện bên nửa trái của khung hình thứ hai $s21[i]$, số lượng phương tiện bên nửa phải của khung hình thứ hai: $s22[i]$. Ta sẽ tính được lưu lượng xe: $L[i] = \max(0, (s22[i] - (s21[i] - s11[i])) - (s22[i] - s12[i]))$. Công thức tính này chỉ mang tính tương đối và so sánh lượng xe giữa hai khung hình và vị trí của chúng.

Tiếp theo, cho trước hai số nguyên $k1$ và $k2$, module lấy trung bình tổng số phương tiện trên t khung hình. Nếu trung bình đó lớn hơn $k1$, thì hiện tại khu vực đó được cho là đang đông xe. Ngược lại thì sẽ là đang vắng xe. Tiếp theo, nếu ta lấy trung bình cộng của các $L[i]$. Nếu trung bình cộng này vượt mức $k2$ thì ta nói lưu lượng xe nhanh. Ngược lại thì lưu lượng xe chậm.

Cuối cùng, ta thực hiện phân loại: Nếu ta có Đông xe, lưu lượng xe chậm thì hiện tại giao thông

đang ùn tắc; nếu ta có Đông xe, lưu lượng xe nhanh thì hiện tại giao thông đang có nguy cơ ùn tắc; nếu ta có Vắng xe, ... thì ta nói hiện tại giao thông đang không ùn tắc.

10.4 Phương pháp đánh giá mô hình

Nhóm dự định sẽ đánh giá mô hình dựa trên hai tiêu chí ứng với hai đầu ra của các đơn vị edge: lưu lượng xe và phân loại tình trạng giao thông.

Định lượng: Nhằm xác định khả năng phát hiện và phân loại đúng các vật thể phương tiện trong dữ liệu từ môi trường, nhóm sẽ nhận về các ảnh đầu vào từ các đơn vị edge, sau đó (1) đếm các phương tiện thủ công và so sánh với đầu ra của các đơn vị edge hoặc (2) sử dụng một mô hình phát hiện phương tiện lớn hơn nhằm tự động hóa cả quá trình.

Nhóm cũng dự định sẽ sử dụng một tập kiểm thử có gắn nhãn nhằm có một quy trình đánh giá đúng đắn hơn trước khi triển khai hệ thống vào thực tiễn.

Định tính: Nhóm sẽ thực hiện kiểm tra thủ công một số nhỏ các mẫu đầu ra của mô hình và so sánh nó với hình ảnh thực tế của tình trạng giao thông vào lúc đó. Nhóm sẽ quyết định phân loại giao thông ùn tắc/không ùn tắc/có nguy cơ ùn tắc dựa vào: **Lưu lượng xe và Tốc độ trung bình** của tất cả xe trên khung hình.

Ngoài ra, để đánh giá tổng quan tính nhất quán của cả hệ thống, các module khác (như lưu trữ dữ liệu cũ, cung cấp giao diện người dùng, phương pháp tái thiết lập kết nối khi có trục trặc) sẽ được nhóm kiểm thử bằng các phương pháp kiểm thử phần mềm thông dụng.

11 Kế hoạch

11.1 Giai đoạn 1

- Ngày bắt đầu: 19/05/2025
- Ngày kết thúc: 01/06/2025
- Công việc:
 - Lập kế hoạch phát triển.
 - Nghiên cứu các giải pháp về mặt phần mềm.
 - Nghiên cứu các giải pháp về mặt phần cứng.
 - Đặc tả yêu cầu bài toán.
 - Đặc tả kiến trúc hệ thống.
 - Đặc tả các luồng sử dụng của hệ thống.

11.2 Giai đoạn 2

- Ngày bắt đầu: 25/07/2025
- Ngày kết thúc: 28/07/2025
- Công việc:
 - Xây dựng hệ thống rút trích đặc trưng.
 - Xây dựng hệ thống dự đoán vị trí.

- Xây dựng hệ thống dự đoán tình trạng giao thông.
- Xây dựng cơ sở dữ liệu.
- Xây dựng giao diện người dùng.
- Dụng máy chủ MQTT.
- Thu các yêu cầu phần cứng.
- Xây dựng phần cứng.
- Kết hợp phần cứng và phần mềm trên đơn vị edge.

11.3 Giai đoạn 3

- Ngày bắt đầu: 29/07/2025
- Ngày kết thúc: 31/07/2025
- Công việc:
 - Kiểm thử cả quy trình chạy dựa trên tập dữ liệu thử.

11.4 Giai đoạn 4

- Ngày bắt đầu: 01/08/2025
- Ngày kết thúc: 02/08/2025
- Công việc:
 - Đánh giá hiệu suất mô hình.
 - Chuẩn bị slide và tài liệu thuyết trình.
 - Thuyết trình sản phẩm.

12 Nguồn tham khảo

- [1] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, *Mcunet: Tiny deep learning on iot devices*, <https://arxiv.org/abs/2007.10319>, NeurIPS, 2020.
- [2] J. Lin, W.-M. Chen, H. Cai, C. Gan, and S. Han, *Mcunetv2: Memory-efficient patch-based inference for tiny deep learning*, <https://arxiv.org/abs/2110.15352>, arXiv preprint arXiv:2110.15352, 2021.
- [3] L. Yang, P. Luo, C. C. Loy, and X. Tang, *A large-scale car dataset for fine-grained categorization and verification*, https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Yang_A_Large-Scale_Car_2015_CVPR_paper.html, In Computer Vision and Pattern Recognition (CVPR), 2015.
- [4] B. Jacob et al., *Quantization and training of neural networks for efficient integer-arithmetic-only inference*, <https://arxiv.org/abs/1712.05877>, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2704–2713, 2018.
- [5] A. G. Howard et al., *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, <https://arxiv.org/abs/1704.04861>, arXiv preprint arXiv:1704.04861, 2017.