BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI CÔNG NGHỆ KỸ THUẬTT
THÀNH PHỐ HỒ CHÍ MINH

Nguyễn Nhựt Đăng
Ngô Trọng Nghĩa

Xây dựng hệ thống mô phỏng dự báo và điều chỉnh đèn giao thông dựa trên dữ liệu snapshot từ camera giao thông sử dụng học sâu và mô phỏng SUMO

KHÓA LUẬN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

TP. Hồ Chí Minh, Tháng 01, 2026

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

Nguyễn Nhựt Đăng
Ngô Trọng Nghĩa

Xây dựng hệ thống mô phỏng dự báo và điều chỉnh đèn giao thông dựa trên dữ liệu snapshot từ camera giao thông sử dụng học sâu và mô phỏng SUMO

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Người hướng dẫn khoa học: TS. Huỳnh Thế Thiện

TP. Hồ Chí Minh, Tháng 01, 2026

width=!,height=!,pages=-, noautoscale=true, width=height=

Lời Cam Đoan

Với tư cách là người thực hiện khóa luận tốt nghiệp này, chúng tôi là Ngô Trọng Nghĩa, mã số sinh viên 21161155 và Nguyễn Nhựt Đăng, mã số sinh viên 21119062 cùng đang theo học ngành Công nghệ kỹ thuật máy tính tại Khoa Điện - Điện tử, Trường Đại học Sư phạm Kỹ thuật TP.HCM. Chúng tôi xin khẳng định đây hoàn toàn là công trình nghiên cứu do chúng tôi sáng tạo. Nội dung và kết quả của khóa luận phản ánh năng lực chuyên môn, kỹ năng nghiên cứu và sự nỗ lực tự thân của chúng tôi, không hề vay mượn hay sao chép từ bất kỳ đồ án, bài báo hay tài liệu nào đã được công bố trước đây mà không trích dẫn nguồn. Chúng tôi cam đoan mọi tài liệu tham khảo được sử dụng đều đã được ghi nhận đầy đủ và chính xác, tuân thủ nghiêm ngặt quy định về trích dẫn của Nhà trường và các chuẩn mực học thuật quốc tế. Chúng tôi đảm bảo tính xác thực, khách quan của thông tin trình bày và khẳng định không có hành vi học thuật không trung thực. Chúng tôi hoàn toàn chịu trách nhiệm về tính nguyên bản của công trình này và chấp nhận mọi hình thức xử lý kỷ luật nếu phát hiện bất kỳ vi phạm nào đối với bản cam kết này.

Người thực hiện tiểu luận
(Ký và ghi rõ họ tên)

Lê Trường Thịnh

Lời Cảm Tạ

Chúng tôi nhận thức sâu sắc rằng việc hoàn thành khóa luận tốt nghiệp này không thể thực hiện được nếu thiếu đi sự đồng hành, tư vấn và hỗ trợ quý báu từ quý Thầy Cô cùng các bạn bè trong ngành Hệ thống nhúng và IoT, Khoa Điện - Điện tử, Trường Đại Học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh. Chúng tôi xin bày tỏ lòng biết ơn chân thành nhất đến tất cả những người đã dành thời gian, công sức góp ý và giúp đỡ chúng tôi trong suốt quá trình thực hiện công trình này. Đặc biệt, sự dẫn dắt của Thầy Huỳnh Thế Thiện đóng vai trò then chốt. Những định hướng chuyên môn, lời khuyên tận tình và lộ trình nghiên cứu Thầy vạch ra ngay từ những bước đi đầu tiên đã ảnh hưởng sâu sắc đến tư duy và cách tiếp cận đề tài của chúng tôi. Dù đã rất cố gắng, khóa luận chắc chắn vẫn còn tồn tại những điểm chưa hoàn thiện. Chúng tôi rất mong nhận được những góp ý thẳng thắn và đánh giá khách quan để có thể nâng cao kiến thức, khắc phục hạn chế và tạo ra những sản phẩm chất lượng hơn trong tương lai. Xin chân thành cảm ơn!

Tóm Tắt

Nghiên cứu này giới thiệu một phương pháp mới dựa trên mô hình học sâu để nhận diện tín hiệu 5G (fifth-generation), còn được gọi là NR (new radio), và LTE (long-term evolution), với trọng tâm là xác định các vùng phổ tần số của tín hiệu được điều chế trong mạng vô tuyến. Phương pháp này nhằm mục đích hỗ trợ việc xây dựng các mạng vô tuyến nhận thức thế hệ tiếp theo. Về mặt lý thuyết, trong quá trình truyền dẫn, các tín hiệu được điều chế thường trở nên khó nhận diện do có dạng sóng mang phức tạp. Để giải quyết vấn đề này, các tín hiệu thu được từ máy thu sẽ được chuyển đổi thành hình ảnh phổ, giúp hiển thị thông tin trực quan hơn bằng cách áp dụng phép biến đổi Fourier thời gian ngắn (short-time Fourier transform - STFT).

Để xác định vùng phổ của tín hiệu 5G và LTE cùng tồn tại trên một hình ảnh phổ, tác giả giới thiệu một phương pháp cảm biến phổ tiên tiến dành cho các mạng không dây thế hệ tiếp theo, sử dụng kiến trúc hai đường dẫn. Mô hình đổi mới này được thiết kế để phân đoạn chính xác các tín hiệu 5G NR và LTE bằng cách xác định nội dung phổ dựa trên tần số và thời gian mà các tín hiệu chiếm dụng. Phương pháp này tích hợp một đường dẫn ngữ cảnh nhằm thu thập thông tin ngữ nghĩa ở mức độ cao, một đường dẫn không gian để bảo toàn các đặc trưng chi tiết về không gian, và một cơ chế hợp nhất đặc trưng mới nhằm kết hợp hiệu quả thông tin từ cả hai đường dẫn. Kiến trúc này có khả năng học tập cả các đặc trưng phổ cục bộ và toàn cục, qua đó nâng cao đáng kể hiệu suất phân đoạn. Kết quả thực nghiệm cho thấy phương pháp này đạt hiệu quả và hiệu suất vượt trội, với một kiến trúc gọn nhẹ chỉ gồm 7 triệu tham số, đạt được độ chính xác toàn cục (global accuracy) là 97.25% và giá trị trung bình của chỉ số giao nhau trên hợp (mean intersection over union – IoU) là 94.76%. Những kết quả này chứng minh rằng đây là một giải pháp đầy hứa hẹn dành cho các hệ thống thông tin không dây thế hệ tiếp

theo.

Mục Lục

Danh sách hình

Danh sách bảng

Danh sách các từ viết tắt

Các từ viết tắt	**Định nghĩa**
None	Không có

Chương 1

TỔNG QUAN

1.1 GIỚI THIỆU

Trong bối cảnh đô thị hóa nhanh chóng và sự gia tăng mật độ phương tiện giao thông, việc quản lý và điều tiết giao thông hiệu quả đã trở thành một trong những thách thức lớn nhất mà các thành phố hiện đại phải đối mặt. Theo báo cáo của Tổ chức Hợp tác và Phát triển Kinh tế (OECD), tắc nghẽn giao thông không chỉ gây ra thiệt hại kinh tế hàng tỷ USD mỗi năm mà còn là nguyên nhân chính dẫn đến ô nhiễm không khí, tiêu thụ nhiên liệu không hiệu quả và giảm chất lượng cuộc sống của người dân [**?**].

Sự phát triển của công nghệ thông tin và truyền thông (ICT) cùng với sự xuất hiện của các hệ thống giao thông thông minh (Intelligent Transportation Systems - ITS) đã mở ra những cơ hội mới trong việc giải quyết các vấn đề giao thông. Đặc biệt, việc ứng dụng các kỹ thuật học máy và học sâu (Deep Learning) vào phân tích dữ liệu giao thông đã cho thấy những kết quả đầy hứa hẹn trong việc dự báo và tối ưu hóa luồng giao thông [**?**].

Hệ thống camera giao thông hiện đại có khả năng thu thập một lượng lớn dữ liệu hình ảnh theo thời gian thực, tạo ra những "snapshot" phản ánh tình trạng giao thông tại các điểm quan sát. Những dữ liệu này, khi được xử lý bằng các thuật toán học sâu tiên tiến, có thể cung cấp thông tin quý giá về mật độ phương tiện, tốc độ di chuyển, và các mẫu hành vi giao thông, từ đó làm cơ sở cho việc dự báo và điều tiết giao thông hiệu quả.

1.2 MỤC TIÊU

Mục tiêu của đề tài là xây dựng một hệ thống tích hợp giữa thu thập dữ liệu ảnh snapshot từ camera giao thông, xử lý bằng các phương pháp học sâu, và mô phỏng trên nền tảng SUMO nhằm dự báo lưu lượng phương tiện và hỗ trợ điều chỉnh tín hiệu đèn giao thông một cách thích ứng. Hệ thống hướng tới việc khai thác dữ liệu hình ảnh để cung cấp thông tin định lượng về tình trạng giao thông, từ đó nâng cao hiệu quả điều khiển nút giao trong bối cảnh giao thông đô thị.

Cụ thể, đề tài tập trung vào các mục tiêu sau:

- Xây dựng quy trình thu thập và xử lý ảnh snapshot từ camera giao thông, chuyển đổi dữ liệu hình ảnh thô thành các thông số giao thông có cấu trúc như lưu lượng xe, mật độ phương tiện và phân bố phương tiện theo làn, thông qua các mô hình học sâu.
- Tích hợp các thông số giao thông trích xuất được vào môi trường mô phỏng SUMO nhằm tái hiện trạng thái giao thông tại nút giao hoặc khu vực nghiên cứu, phục vụ cho việc phân tích và đánh giá hoạt động của hệ thống tín hiệu đèn.
- Phát triển mô hình dự báo ngắn hạn lưu lượng phương tiện dựa trên chuỗi dữ liệu thời gian thu được từ camera và kết quả mô phỏng, với mục tiêu ước lượng xu hướng biến động giao thông trong các khoảng thời gian kế tiếp.
- Thiết kế và đánh giá chiến lược điều chỉnh tín hiệu đèn giao thông dựa trên kết quả dự báo lưu lượng, nhằm tối ưu hóa các chỉ tiêu vận hành như thời gian chờ trung bình tại nút giao, số lần dừng xe và mức độ ùn tắc.
- Đánh giá hiệu quả của hệ thống thông qua các thước đo như sai số dự báo lưu lượng, thời gian chờ trung bình, chiều dài hàng đợi và khả năng áp dụng trong thực tế, từ đó đề xuất hướng cải tiến và mở rộng cho các hệ thống điều khiển tín hiệu giao thông thông minh.

Tóm lại, mục tiêu của đề tài là xây dựng một công cụ hỗ trợ thông minh cho công tác quản lý và điều khiển tín hiệu giao thông, kết hợp giữa các phương pháp học sâu

và mô phỏng giao thông. Hệ thống cho phép chuyển hóa dữ liệu hình ảnh snapshot từ camera giao thông thành các thông tin định lượng về lưu lượng phương tiện, phục vụ cho bài toán dự báo và điều chỉnh tín hiệu đèn. Qua đó, đề tài hướng tới việc cải thiện hiệu quả vận hành tại các nút giao, giảm thời gian chờ và mức độ ùn tắc, góp phần nâng cao chất lượng dịch vụ giao thông đô thị.

1.3 PHƯƠNG PHÁP NGHIÊN CỨU

Để đạt được các mục tiêu đã đề ra, Nghiên cứu này áp dụng một quy trình nghiên cứu khoa học chặt chẽ, kết hợp giữa nghiên cứu lý thuyết, thực nghiệm mô phỏng và phân tích đánh giá định lượng. Các phương pháp cụ thể được triển khai như sau:

Phương pháp nghiên cứu lý thuyết:

Trước hết, đề tài tiến hành khảo sát, tổng hợp và phân tích các cơ sở lý thuyết và công trình nghiên cứu liên quan đến bài toán giao thông đô thị và các phương pháp tiếp cận hiện đại trong lĩnh vực thị giác máy tính, học sâu và mô phỏng giao thông. Nội dung nghiên cứu tập trung vào các khái niệm nền tảng về hệ thống giao thông, đặc trưng dòng xe, cũng như các chỉ số thường được sử dụng để đánh giá trạng thái giao thông như lưu lượng, mật độ và thời gian chờ.

Tiếp theo, đề tài nghiên cứu các mô hình học sâu phục vụ nhận dạng và đếm phương tiện từ dữ liệu hình ảnh, trong đó tập trung vào các mô hình phát hiện đối tượng một giai đoạn như YOLO, đặc biệt là YOLOv11, đã được chứng minh hiệu quả trong các bài toán đếm xe và ước lượng lưu lượng giao thông từ camera giám sát. Song song với đó, các mô hình dự báo chuỗi thời gian, tiêu biểu là Long Short-Term Memory (LSTM), được khảo sát nhằm khai thác mối quan hệ theo thời gian của dữ liệu lưu lượng, phục vụ cho bài toán dự báo ngắn hạn.

Bên cạnh đó, đề tài tổng hợp các nghiên cứu liên quan đến mô phỏng giao thông vi mô bằng SUMO, cũng như các hướng tiếp cận tích hợp giữa dữ liệu thu thập từ camera, kết quả nhận dạng phương tiện và mô hình mô phỏng. Các công trình về điều khiển tín hiệu giao thông dựa trên dữ liệu mô phỏng và dự báo cũng được phân tích nhằm làm rõ

cách thức ánh xạ từ dữ liệu quan sát sang các tham số điều khiển trong môi trường mô phỏng. Từ những cơ sở lý thuyết này, đề tài xây dựng nền tảng cho việc lựa chọn kiến trúc hệ thống, xác định các tham số mô hình và đề xuất các tiêu chí đánh giá hiệu quả như độ chính xác dự báo, lưu lượng, mật độ phương tiện và thời gian chờ tại nút giao.

Phương pháp thực nghiệm mô phỏng:

- **Thu thập dữ liệu snapshot camera và xử lý bằng học sâu:** Sử dụng mô hình YOLOv11 để phát hiện và đếm phương tiện từ ảnh hoặc video snapshot thu được từ camera giao thông. Từ kết quả đếm được xác định các thông số như lưu lượng xe, mật độ theo từng mốc thời gian và từng điểm giao thông. Đây chính là bước chuyển dữ liệu hình ảnh thô thành dạng dữ liệu có cấu trúc để sử dụng tiếp. Việc này tận dụng các thư viện, công cụ thực nghiệm đã có (ví dụ từ thực tế/nguồn mở) như một số nghiên cứu đã thực hiện.

- **Xây dựng mô hình dự báo mật độ giao thông:** Từ chuỗi dữ liệu mật độ theo mốc thời gian thu được, tiến hành huấn luyện mô hình LSTM để dự báo mật độ giao thông cho 15 phút tiếp theo. Việc này gồm tiền xử lý dữ liệu (chuỗi thời gian, tạo các đặc trưng như thời gian, ngày, giờ, điểm giao thông, loại phương tiện nếu có), chia train/validation/test, lựa chọn cấu trúc mạng LSTM (số lớp, số đơn vị, dropout, epochs...), và kiểm định mô hình qua các chỉ tiêu như RMSE, MAE, MAPE. (?)

- **Tích hợp mô phỏng giao thông với SUMO:** Sử dụng môi trường SUMO để tái hiện mạng lưới giao thông nghiên cứu, cấu hình các tham số như làn đường, tín hiệu giao thông, các điểm camera/tuyến đường tương ứng với dữ liệu thực tế. Sau đó, nhập dữ liệu mật độ xe thực tế (hoặc dữ liệu từ mô hình đếm) để khớp mô phỏng sao cho trạng thái mô phỏng càng sát thực càng tốt. Đây là bước "đồng bộ" giữa dữ liệu thực và mô phỏng.

- **Tích hợp module điều khiển giao thông:** Xây dựng thuật toán điều tiết giao thông (ví dụ: điều chỉnh tín hiệu, phân luồng, ưu tiên làn) dựa trên dự báo mật độ từ LSTM và kết quả mô phỏng. Thử chạy hai kịch bản: kịch bản không điều tiết (tín hiệu cố định hoặc theo chế độ hiện hữu) và kịch bản có điều tiết (tín hiệu và

phân luồng thay đổi dựa trên dự báo và mô phỏng). Chạy mô phỏng SUMO cho cả hai kịch bản và thu thập dữ liệu kết quả.

- **Xây dựng dashboard trực quan:** Triển khai giao diện hiển thị gồm luồng xe chạy như mô phỏng (trong SUMO), biểu đồ mật độ theo thời gian, kết quả dự báo LSTM, và so sánh hiệu quả giữa hai kịch bản (có/không điều tiết). Dashboard sẽ giúp trực quan hóa kết quả và hỗ trợ phân tích.

- **Rút ra kết luận, đề xuất cải tiến và kiến nghị ứng dụng thực tiễn:** từ kết quả thực nghiệm và mô phỏng, chỉ rõ giới hạn của nghiên cứu, gợi ý mở rộng (ví dụ mở rộng mạng lưới, loại phương tiện đa dạng, tích hợp dữ liệu thời tiết, sự cố...).

Phương pháp phân tích đánh giá:

- **Đánh giá độ chính xác của khối phát hiện và đếm phương tiện từ camera:** Kết quả đếm phương tiện tự động được đối chiếu với số liệu đếm thủ công hoặc các nguồn dữ liệu tham chiếu (nếu có), nhằm xác định mức độ chính xác và độ tin cậy của mô hình trong các điều kiện giao thông khác nhau.

- **Phân tích hiệu năng dự báo của mô hình LSTM:** Hiệu quả dự báo mật độ phương tiện trong khoảng thời gian 15 phút tiếp theo được đánh giá thông qua các chỉ tiêu định lượng phổ biến như RMSE, MAE, MAPE và hệ số xác định R^2, từ đó phản ánh khả năng nắm bắt xu hướng và biến động của lưu lượng giao thông.

- **Phân tích kết quả mô phỏng giao thông bằng SUMO:** So sánh các kịch bản không áp dụng điều tiết và có áp dụng điều tiết đèn tín hiệu trên các tiêu chí như thời gian lưu thông trung bình, mật độ phương tiện, số lần dừng/chờ và mức độ ổn định của luồng giao thông, nhằm đánh giá hiệu quả của chiến lược điều khiển.

- **Đánh giá tổng thể hệ thống:** Phân tích mức độ tích hợp và phối hợp giữa các thành phần của hệ thống (phát hiện–đếm phương tiện, dự báo lưu lượng, mô phỏng giao thông và điều chỉnh đèn tín hiệu), đồng thời xem xét tính khả thi khi triển khai trong thực tế và khả năng thích ứng trước các biến động giao thông như giờ cao điểm hoặc sự cố bất thường.

- **Trực quan hóa và phân tích kết quả trên dashboard:** Trình bày và so sánh các biểu đồ, luồng giao thông và chỉ số vận hành trước và sau điều chỉnh đèn tín hiệu, từ đó rút ra nhận xét về hiệu quả điều tiết và lợi ích mang lại cho công tác quản lý giao thông đô thị.

1.4 GIỚI HẠN NGHIÊN CỨU

Mặc dù hệ thống nghiên cứu đã nỗ lực tích hợp các thành phần thu thập dữ liệu hình ảnh, học sâu, dự báo lưu lượng và mô phỏng điều tiết giao thông thành một quy trình tương đối hoàn chỉnh, song vẫn còn tồn tại một số hạn chế cần được xem xét. Trước hết, việc sử dụng ảnh snapshot từ camera giao thông chỉ phản ánh trạng thái giao thông tại những thời điểm rời rạc, mang tính "tĩnh", và chịu ảnh hưởng đáng kể từ các điều kiện quan sát như góc đặt camera, điều kiện chiếu sáng, mức độ che khuất và chất lượng hình ảnh. Do đó, hiệu quả phát hiện và đếm phương tiện của mô hình YOLOv11 có thể suy giảm trong các điều kiện môi trường bất lợi (mưa, sương mù, bóng đổ) hoặc tại các khu vực giao thông phức tạp, nơi mật độ phương tiện cao và làn đường không được phân định rõ ràng.

Bên cạnh đó, dữ liệu thu thập tại một hoặc một số vị trí giao thông cụ thể chưa đủ để phản ánh đầy đủ đặc trưng của toàn bộ mạng lưới giao thông đô thị cũng như sự đa dạng của các điều kiện vận hành theo thời gian (giờ cao điểm, ngày lễ, hoặc các tình huống sự cố). Điều này có thể làm hạn chế khả năng khái quát hóa kết quả dự báo và mô phỏng sang các bối cảnh giao thông khác.

Ngoài ra, do hạn chế về kinh phí, nghiên cứu chỉ có thể khai thác nguồn dữ liệu ảnh snapshot từ các camera giao thông công khai, vốn có vị trí lắp đặt cố định và không thể điều chỉnh góc quay. Hệ quả là nhiều ảnh thu được có góc nhìn không thuận lợi (góc quay hẹp, góc xiên), bị che khuất bởi các vật cản như cây cối, biển báo hoặc cột đèn, cũng như có chất lượng hình ảnh không đồng đều. Những yếu tố này làm giảm độ chính xác của quá trình nhận diện và đếm phương tiện, từ đó dẫn đến sai số trong việc ước lượng mật độ giao thông và ảnh hưởng đến các bước phân tích tiếp theo của hệ thống.

Về phần mô hình học sâu và dự báo bằng Long Short-Term Memory (LSTM), mặc

dù có khả năng dự báo mật độ giao thông cho 15 phút tiếp theo, nhưng việc huấn luyện chỉ trên dữ liệu từ các mốc thời gian cố định và điểm giao thông nhất định khiến mô hình có thể kém chính xác khi đối mặt với tình huống bất thường (như tai nạn, thay đổi bất ngờ lưu lượng, thời tiết cực đoan) mà không có dữ liệu học trước. Hơn nữa, mô hình dự báo chỉ tập trung vào một biến chính - mật độ phương tiện - và chưa tích hợp đầy đủ các yếu tố khác như loại phương tiện, tốc độ, hành vi người lái, ảnh hưởng từ tín hiệu đèn hoặc thay đổi hành lang giao thông.

Phần mô phỏng giao thông với SUMO dù được "khớp" với dữ liệu thực tế đến mức có thể nhưng vẫn có giới hạn vì mô phỏng luôn là bản sao không hoàn hảo của môi trường thực. Mạng lưới mô phỏng có thể chưa mô hình hóa đầy đủ mọi chiều của thực tế như sự tương tác phức tạp giữa các phương tiện, hành vi bất định, ảnh hưởng thời tiết, người đi bộ, xe máy nhỏ, hoặc việc vi phạm giao thông - những yếu tố rất phổ biến tại đô thị như Hồ Chí Minh. Việc điều tiết giao thông thông qua thuật toán cũng đặt giả định rằng các thông số mô hình và dữ liệu đầu vào là đồng nhất và ổn định, trong khi thực tế có thể thay đổi nhanh và không thể đo trước hết.

Cuối cùng, việc xây dựng dashboard trực quan để hiển thị kết quả mô phỏng, dự báo và điều tiết cũng gặp giới hạn do khả năng phản ánh toàn bộ thực tế - dashboard chỉ hiển thị dữ liệu và mô phỏng ở mức độ "có thể" và phù hợp với giả định nghiên cứu. Những quyết định điều tiết đưa ra từ mô hình có thể chưa tính tới đầy đủ chi phí thực thi, điều kiện vận hành thực tế, phản ứng của người tham giao thông hoặc các yếu tố tổ chức giao thông ngoài mô hình.

Tóm lại, các giới hạn nghiên cứu chính bao gồm: dữ liệu nguồn (chỉ snapshot camera, tại các điểm giới hạn), khả năng khái quát hóa kết quả mô hình và mô phỏng, độ chính xác và phạm vi của mô hình dự báo, sự đơn giản hóa môi trường mô phỏng và giả định điều tiết, và khả năng ứng dụng thực tế bị chi phối bởi nhiều yếu tố ngoài mô hình. Hiểu và thừa nhận các giới hạn này giúp bạn đọc đánh giá đúng mức độ đóng góp của nghiên cứu, và tạo nền tảng cho các nghiên cứu tiếp theo.

1.5 BỐ CỤC

Chương 1: Tổng quan - Trình bày bối cảnh, động lực nghiên cứu, mục tiêu và phương pháp nghiên cứu.

Chương 2: Cơ sở lý thuyết - Tổng quan các kiến thức nền tảng về xử lý ảnh, học sâu, dự báo chuỗi thời gian và học tăng cường.

Chương 3: Thiết kế hệ thống - Trình bày kiến trúc tổng thể của hệ thống và thiết kế chi tiết các mô-đun.

Chương 4: Kết quả vào thảo luận - Mô tả quá trình triển khai hệ thống và các thí nghiệm đánh giá.

Chương 5: Kết luận và hướng phát triển - Tổng kết những đóng góp của nghiên cứu và đề xuất hướng phát triển tương lai.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH

Xử lý ảnh (Image Processing) là lĩnh vực thuộc khoa học máy tính và kỹ thuật, tập trung vào việc phân tích và biến đổi hình ảnh nhằm cải thiện chất lượng hoặc trích xuất thông tin phục vụ quan sát và nhận dạng. Đây là nền tảng quan trọng của thị giác máy tính, vì hầu hết các thuật toán phân tích hay học máy đều yêu cầu dữ liệu hình ảnh đã được xử lý chuẩn hóa. [**?**]

Về bản chất, xử lý ảnh là quá trình thao tác trực tiếp trên ma trận điểm ảnh để làm rõ thông tin hữu ích và loại bỏ nhiễu hay các thành phần không cần thiết. Các kỹ thuật này thường được chia làm hai mức: xử lý ảnh mức thấp và mức cao. Mức thấp chủ yếu gồm các phép biến đổi tín hiệu như lọc nhiễu, điều chỉnh độ sáng, thay đổi không gian màu hoặc tăng độ sắc nét mà không xét đến nội dung ảnh. Trong khi đó, xử lý ảnh mức cao tập trung vào việc trích xuất đặc trưng như biên cạnh, điểm đặc trưng, kết cấu hay hình dạng, tạo dữ liệu có ý nghĩa cho các mô hình nhận diện và học sâu. Các thành phần cơ bản của xử lý ảnh:

- **Hệ thống thu nhận hình ảnh (Image Sensor):** Để thu được ảnh số, cần hai thành phần cơ bản. Thứ nhất là cảm biến vật lý, có khả năng phản hồi với năng lượng phát ra từ đối tượng quan sát. Thứ hai là thiết bị chuyển đổi tín hiệu từ cảm biến sang dạng số, thường gọi là bộ số hóa (digitizer). Bộ số hóa này chịu trách

nhiệm biến các tín hiệu thu được từ cảm biến thành dữ liệu số để máy tính có thể xử lý.

- **Phần cứng xử lý ảnh chuyên dụng (Specialized Image Processing Hardware):** Để thực hiện các phép tính số học và logic trên toàn bộ ảnh, cần kết hợp giữa bộ số hóa và phần cứng chuyên dụng, thường được gọi là hệ thống tiền xử lý (front-end subsystem). Tốc độ xử lý của phần cứng này là yếu tố quan trọng nhất, vì các máy tính thông thường khó đáp ứng được yêu cầu truyền dữ liệu với tốc độ cao cần thiết cho xử lý ảnh thời gian thực.

- **Máy tính: (Computer):** Máy tính trong hệ thống xử lý ảnh là máy tính đa năng, có thể là một PC thông thường hoặc siêu máy tính tùy vào quy mô ứng dụng. Một máy tính cá nhân cấu hình tốt thường đủ cho các tác vụ xử lý ảnh offline, phục vụ nghiên cứu và phân tích dữ liệu hình ảnh.

- **Phần mềm xử lý ảnh (Image Processing Software):** Phần mềm xử lý ảnh gồm các module chuyên dụng thực hiện những nhiệm vụ cụ thể. Một bộ phần mềm thiết kế tốt sẽ cho phép người dùng viết ít lệnh nhất, đồng thời tận dụng tối đa các module có sẵn. Các gói phần mềm phát triển cao còn cho phép tích hợp các module và câu lệnh lập trình từ ít nhất một ngôn ngữ lập trình. Ví dụ, MATLAB là một trong những công cụ phổ biến được dùng trong các hệ thống xử lý ảnh.

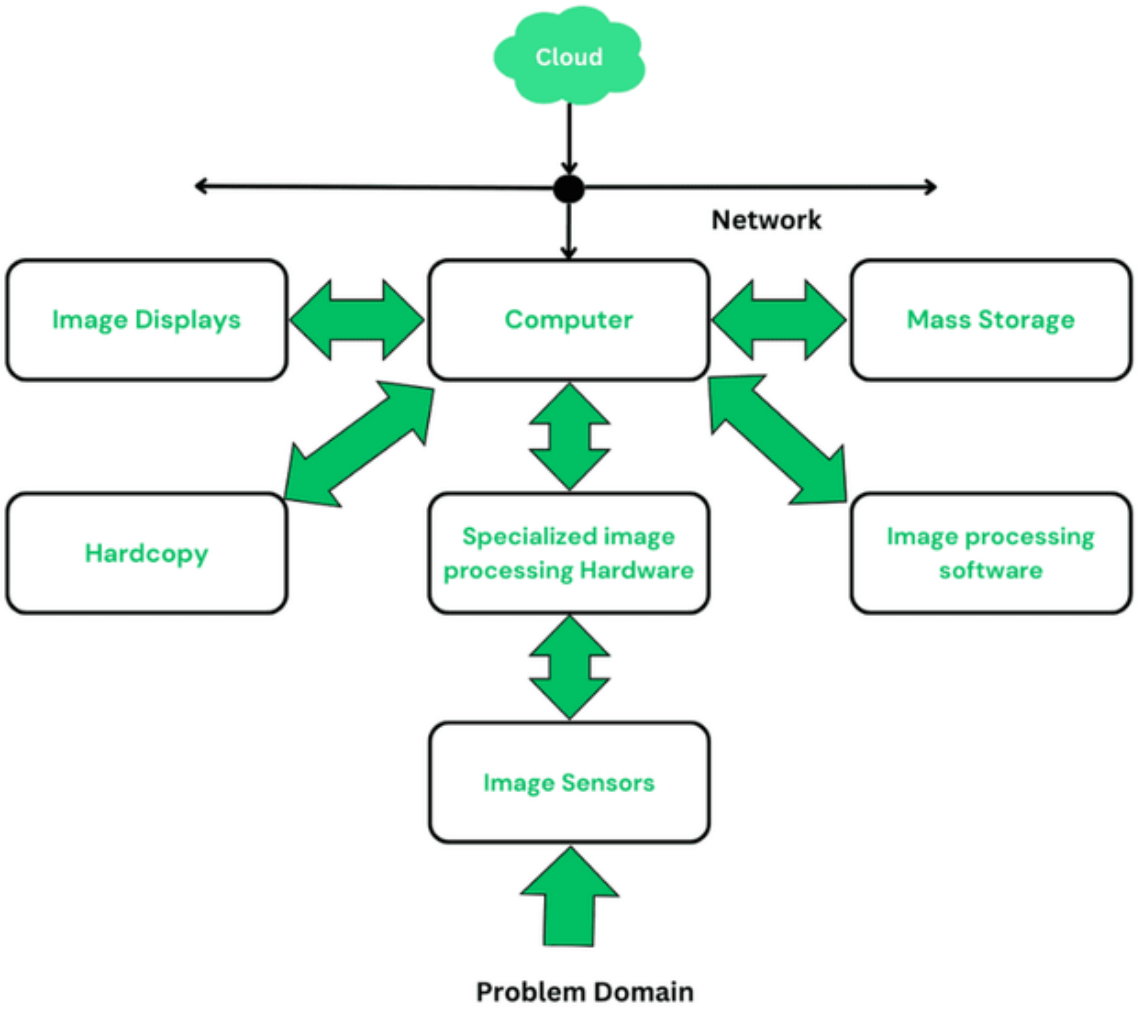
- **Lưu trữ dữ liệu (Mass Storage):** Lưu trữ là yếu tố quan trọng trong xử lý ảnh, đặc biệt khi làm việc với các ảnh có dung lượng lớn. Ví dụ, một ảnh có kích thước 1024 x 1024 pixel cần khoảng 1 megabyte nếu chưa nén. Các hệ thống xử lý ảnh thường phải lưu trữ hàng nghìn hoặc thậm chí hàng triệu ảnh. Hệ thống lưu trữ số tuân theo ba nguyên tắc cơ bản: Lưu trữ tạm thời (dùng trong quá trình xử lý), lưu trữ trực tuyến (phục vụ truy suất nhanh) và lưu trữ lâu dài (truy xuất ít, lưu trữ dài hạn.).

- **Hiển thị hình ảnh (Image Displays):** Màn hình hiển thị thường là màn hình màu phẳng, được điều khiển bởi card đồ họa hoặc card hiển thị hình ảnh. Đây là thành phần quan trọng giúp máy tính trình chiếu và thao tác với dữ liệu hình ảnh.

- **Thiết bị in ấn (Hardcopy):** Để lưu trữ hoặc trình bày hình ảnh, có thể dùng các thiết bị in ấn và ghi hình, bao gồm máy in laser, máy ảnh phim, thiết bị nhạy

nhiệt, máy in phun, hoặc các phương tiện số như ổ đĩa quang và CD-ROM. Phim ảnh cung cấp độ phân giải cao nhất, trong khi giấy là phương tiện dễ sử dụng để trình bày nội dung. Khi dùng thiết bị chiếu ảnh số, hình ảnh vẫn tồn tại dưới dạng dữ liệu số, giúp dễ dàng trình chiếu hoặc lưu trữ lâu dài.

- **Mạng và điện toán đám mây (Cloud):** Trong thời đại hiện nay, mạng và điện toán đám mây là những yếu tố thiết yếu trong xử lý ảnh. Vì dữ liệu hình ảnh thường có dung lượng rất lớn, băng thông trở thành vấn đề quan trọng khi truyền tải. Khi gửi dữ liệu qua Internet đến các địa điểm từ xa, hiệu quả truyền tải không phải lúc nào cũng cao, do đó công nghệ cáp quang và các giải pháp băng thông rộng được sử dụng. Bên cạnh đó, nén dữ liệu ảnh đóng vai trò quan trọng để giảm dung lượng truyền tải, giúp gửi lượng lớn hình ảnh một cách nhanh chóng và hiệu quả. [?]



Hình 2.1.1: Các thành phần cơ bản của xử lý ảnh

Trong bối cảnh ứng dụng hiện đại, đặc biệt là các hệ thống thông minh như theo dõi giao thông, giám sát đô thị hay phân tích dữ liệu từ camera, xử lý ảnh giữ vai trò như một giai đoạn tiền xử lý không thể thiếu. Ví dụ, dữ liệu từ camera giao thông thường gặp nhiều hạn chế: ánh sáng thay đổi liên tục, điều kiện thời tiết gây nhiễu, độ phân giải không đồng đều, và vật thể thường nhỏ hoặc bị che khuất. Nếu không có bước xử lý ảnh phù hợp, những hạn chế này sẽ ảnh hưởng trực tiếp đến độ chính xác của các mô hình phát hiện và đếm phương tiện. Các kỹ thuật phổ biến như cân bằng histogram, lọc Gaussian, chuyển đổi sang ảnh xám, khử nhiễu bằng median filter hay tăng độ tương phản đều được áp dụng để cải thiện độ rõ ràng trước khi đưa ảnh vào pipeline phân tích.

Sự phát triển của học sâu trong hơn một thập kỷ qua đã mở ra một hướng mới cho xử lý ảnh, nơi mà các mô hình không chỉ thực hiện các phép biến đổi dựa trên quy tắc cố định mà còn học trực tiếp từ dữ liệu để tối ưu hóa chất lượng đầu ra. Các bài toán như khử nhiễu (denoising), tăng độ phân giải (super-resolution), tái tạo ảnh thiếu thông tin (inpainting), tách nền và nhiều dạng biến đổi phức tạp khác đều đạt được chất lượng vượt trội nhờ mạng nơ-ron tích chập (CNN) và các mô hình sinh ảnh như GAN. Đặc biệt, các hệ thống giám sát giao thông sử dụng camera có thể hưởng lợi trực tiếp từ những kỹ thuật này: ảnh từ camera độ phân giải thấp có thể được tăng cường bằng super-resolution, giúp mô hình phát hiện phương tiện hoạt động chính xác hơn trong môi trường phức tạp. [**?**]

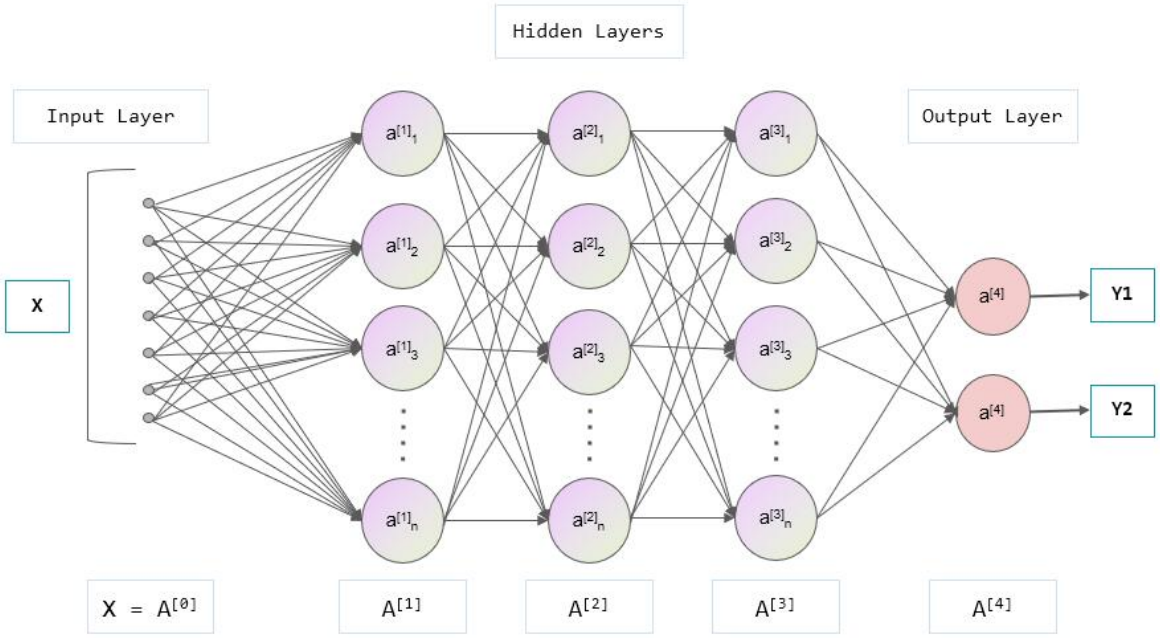
Ngoài ra, xử lý ảnh còn liên quan chặt chẽ đến việc chuẩn hóa dữ liệu đầu vào cho các thuật toán học máy. Việc thay đổi kích thước, chuẩn hóa pixel theo phân phối thống nhất, hoặc điều chỉnh tỷ lệ khung hình đều giúp giảm tải tính toán và tăng độ ổn định khi huấn luyện mô hình nhận dạng hoặc dự đoán. Điều này đặc biệt quan trọng trong các hệ thống thời gian thực, nơi tốc độ xử lý và độ ổn định đóng vai trò quyết định.

Tóm lại, xử lý ảnh là bước khởi đầu cho mọi đề tài thị giác máy tính, cung cấp nền tảng và dữ liệu chất lượng cho các thuật toán học sâu, phát hiện đối tượng và phân tích hành vi. Với sự phát triển nhanh chóng của các mô hình hiện đại, xử lý ảnh không chỉ còn dừng lại ở các kỹ thuật truyền thống mà đang chuyển mình mạnh mẽ và đóng vai trò quan trọng trong việc xây dựng những hệ thống thông minh, chính xác và tin cậy — đặc biệt trong lĩnh vực mô phỏng và quản lý giao thông dựa trên dữ liệu từ camera.

2.2 TỔNG QUAN VỀ LÝ THUYẾT HỌC SÂU

Học sâu (Deep Learning) là một nhánh phát triển nâng cao của học máy (Machine Learning), lấy cảm hứng từ cấu trúc và cơ chế hoạt động của hệ thần kinh sinh học. Trọng tâm của học sâu là các mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN) với nhiều tầng xử lý, cho phép mô hình học được các biểu diễn dữ liệu phức tạp thông qua việc xếp chồng nhiều phép biến đổi phi tuyến. Nhờ khả năng học đặc trưng tự động từ dữ liệu thô, học sâu đã trở thành nền tảng cốt lõi của nhiều hệ thống trí tuệ nhân tạo hiện đại.

Mạng nơ-ron nhân tạo (ANN) là mô hình cơ bản nhất trong học sâu, bao gồm các nút nơ-ron được liên kết với nhau thông qua các trọng số. Khi số lượng lớp ẩn trong ANN tăng lên, mô hình được gọi là mạng nơ-ron sâu (Deep Neural Network - DNN). DNN có khả năng xấp xỉ các hàm phi tuyến phức tạp và mô hình hóa các mối quan hệ đa chiều trong dữ liệu lớn. Quá trình huấn luyện DNN được thực hiện thông qua thuật toán lan truyền ngược (backpropagation) kết hợp với các phương pháp tối ưu, cho phép điều chỉnh trọng số sao cho hàm mất mát được tối thiểu hóa.



Hình 2.2.2: Sơ đồ kiến trúc mạng nơ-ron sâu (Deep Neural Network) bao gồm lớp đầu vào, 3 lớp ẩn và lớp đầu ra.

Trong lĩnh vực xử lý ảnh và thị giác máy tính, mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đóng vai trò trung tâm. CNN được thiết kế chuyên biệt để khai thác cấu trúc không gian của dữ liệu hình ảnh thông qua các lớp tích chập và pooling. Cơ chế này giúp mô hình học được các đặc trưng từ mức thấp (cạnh, góc) đến mức cao (hình dạng, đối tượng), đồng thời giảm đáng kể số lượng tham số so với DNN truyền thống. Nhờ đó, CNN trở thành nền tảng của nhiều mô hình phát hiện và nhận dạng đối tượng hiện đại, tiêu biểu như các phiên bản YOLO được sử dụng rộng rãi trong các hệ thống giao thông thông minh.

Bên cạnh dữ liệu dạng không gian, nhiều bài toán thực tế liên quan đến dữ liệu chuỗi theo thời gian, nơi thứ tự và mối quan hệ phụ thuộc theo thời gian đóng vai trò quan trọng. Mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) được đề xuất nhằm xử lý loại dữ liệu này bằng cách cho phép thông tin từ các bước thời gian trước ảnh hưởng đến quá trình xử lý hiện tại. Tuy nhiên, RNN truyền thống gặp hạn chế trong việc học các phụ thuộc dài hạn do hiện tượng tiêu biến hoặc bùng nổ gradient.

Để khắc phục nhược điểm đó, Long Short-Term Memory (LSTM) - một biến thể của RNN - đã được phát triển với cơ chế cổng (gates) nhằm kiểm soát việc ghi nhớ và quên thông tin theo thời gian. Nhờ khả năng nắm bắt tốt các mối quan hệ dài hạn trong chuỗi dữ liệu, LSTM được ứng dụng hiệu quả trong các bài toán dự báo chuỗi thời gian, chẳng hạn như dự báo lưu lượng giao thông, mật độ phương tiện hoặc xu hướng biến đổi trạng thái giao thông theo thời gian.

Tóm lại, học sâu cung cấp một khung lý thuyết toàn diện cho việc xử lý và phân tích dữ liệu phức tạp ở nhiều dạng khác nhau, từ dữ liệu hình ảnh đến dữ liệu chuỗi thời gian. Phần tiếp theo trình bày khái quát các cơ sở lý thuyết nền tảng của một số mô hình học sâu đã được nghiên cứu và có mối liên hệ trực tiếp với nội dung của đề tài.

2.2.1 Mạng nơ-ron sâu

Mạng nơ-ron sâu (Deep Neural Networks - DNN) là một nhánh quan trọng của học sâu, được phát triển dựa trên nền tảng của mạng nơ-ron nhân tạo truyền thống (Artificial Neural Networks - ANN). Điểm khác biệt cốt lõi của DNN so với các mô hình nơ-ron cổ điển nằm ở số lượng lớn các tầng ẩn (hidden layers), cho phép mô hình học được những

biểu diễn đặc trưng có mức trừu tượng cao từ dữ liệu đầu vào. Nhờ khả năng biểu diễn mạnh mẽ này, DNN đã trở thành nền tảng cho nhiều mô hình học sâu hiện đại và đạt được những thành tựu vượt trội trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên và phân tích chuỗi thời gian.

Về mặt cấu trúc, một mạng nơ-ron sâu thường bao gồm ba thành phần chính: Tầng đầu vào (input layer), nhiều tầng ẩn (hidden layers) và tầng đầu ra (output layer). Mỗi tầng được cấu thành từ các nơ-ron, trong đó mỗi nơ-ron thực hiện phép biến đổi tuyến tính trên dữ liệu đầu vào thông qua các trọng số (weights) và hệ số dịch (bias), sau đó áp dụng một hàm kích hoạt phi tuyến (activation function) như ReLU, sigmoid hoặc tanh. Việc kết hợp nhiều tầng phi tuyến cho phép DNN xấp xỉ các hàm phức tạp, từ đó mô hình hóa được mối quan hệ phi tuyến trong dữ liệu thực tế.

Quá trình huấn luyện mạng nơ-ron sâu được thực hiện thông qua thuật toán lan truyền ngược sai số (backpropagation) kết hợp với các phương pháp tối ưu hóa như Gradient Descent, Adam hoặc RMSProp. Trong quá trình này, sai số giữa đầu ra dự đoán của mô hình và giá trị mục tiêu được lan truyền ngược từ tầng đầu ra về các tầng trước để cập nhật trọng số, nhằm giảm hàm mất mát (loss function). Tuy nhiên, khi số lượng tầng tăng lên, DNN có thể gặp phải các vấn đề như hiện tượng tiêu biến gradient (vanishing gradient) hoặc bùng nổ gradient (exploding gradient), đòi hỏi các kỹ thuật cải tiến như chuẩn hóa dữ liệu, khởi tạo trọng số hợp lý hoặc sử dụng các kiến trúc mạng tiên tiến hơn.

Trong thực tế, DNN đóng vai trò là nền tảng chung cho nhiều kiến trúc học sâu chuyên biệt. Các mô hình như mạng nơ-ron tích chập (CNN) được thiết kế để xử lý dữ liệu dạng ảnh và không gian, trong khi mạng nơ-ron hồi tiếp (RNN) và các biến thể như LSTM phù hợp với dữ liệu chuỗi thời gian. Trong phạm vi đề tài này, DNN là cơ sở lý thuyết quan trọng cho việc hiểu và triển khai các mô hình phát hiện phương tiện từ ảnh camera giao thông cũng như mô hình dự báo lưu lượng giao thông dựa trên chuỗi dữ liệu theo thời gian.

2.2.2 Mạng nơ-ron tích chập

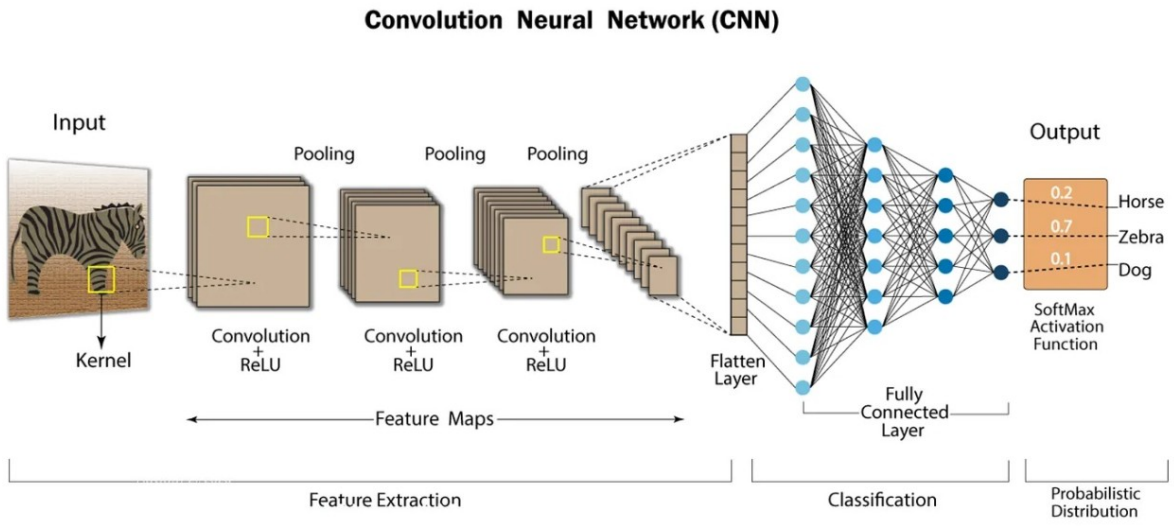
Mạng nơ-ron tích chập (CNN) là kiến trúc mạng sâu tối ưu cho việc xử lý dữ liệu có cấu trúc lưới, đặc biệt là hình ảnh và chuỗi thời gian. Điểm ưu việt của CNN nằm ở cơ chế tích chập, cho phép tự động trích xuất các đặc trưng không gian và cục bộ từ dữ liệu đầu vào một cách hiệu quả. Chính khả năng này đã tạo tiền đề cho những bước tiến vượt bậc trong lĩnh vực thị giác máy tính, đóng vai trò cốt lõi trong các bài toán nhận diện hình ảnh, phát hiện vật thể và phân đoạn ảnh.

Kiến trúc của một mạng CNN điển hình được cấu thành từ ba tầng chức năng chính:

- **Lớp tích chập (Convolutional Layer):** Được coi là "trái tim" của mạng, lớp này chịu trách nhiệm trích xuất đặc trưng thông qua cơ chế tích chập giữa dữ liệu đầu vào và các bộ lọc (filters). Mỗi bộ lọc đóng vai trò như một công cụ dò tìm, học cách nhận diện từ các đường nét cơ bản đến những mẫu hình phức tạp. Đầu ra của quá trình này là các bản đồ đặc trưng (feature maps), chứa đựng các thông tin không gian quan trọng đã được chắt lọc.

- **Lớp gộp (Pooling Layer):** Đóng vai trò tinh gọn mô hình, lớp gộp thực hiện giảm chiều dữ liệu (down-sampling) nhằm giảm thiểu số lượng tham số và khối lượng tính toán, đồng thời hạn chế hiện tượng quá khớp (overfitting). Hai kỹ thuật phổ biến nhất là gộp cực đại (Max Pooling) - lấy giá trị nổi bật nhất, và gộp trung bình (Average Pooling) - tính toán giá trị trung hòa trong vùng lân cận.

- **Lớp kết nối đầy đủ (Fully Connected Layer):** Là chốt chặn cuối cùng của quy trình. Sau khi đi qua các tầng trích xuất và gộp, dữ liệu được duỗi phẳng (flatten) thành vector và đưa vào lớp FC. Tại đây, mạng lưới sẽ tổng hợp toàn bộ thông tin cục bộ để đưa ra quyết định phân loại cuối cùng, thường dưới dạng xác suất dự đoán.



Hình 2.2.3: Sơ đồ kiến trúc mạng nơ-ron tích chập (CNN) bao gồm lớp đầu vào, các lớp tích chập, lớp gộp và lớp kết nối đầy đủ. Các lớp tích chập và gộp đóng vai trò trích xuất và biểu diễn các đặc trưng từ hình ảnh, tạo thành các bản đồ đặc trưng. Sau đó, các bản đồ đặc trưng này được làm phẳng và chuyển qua các lớp kết nối đầy đủ để tạo ra kết quả dự đoán cuối cùng.

Như hình **??** minh họa, quy trình được chia thành hai giai đoạn chính: Trích xuất đặc trưng (Feature Extraction), nơi ảnh đầu vào đi qua các lớp Tích chập (Convolution) kết hợp hàm ReLU và lớp Gộp (Pooling) để tạo ra các bản đồ đặc trưng; và Phân loại (Classification), nơi dữ liệu được làm phẳng (Flatten) và đưa qua các lớp Kết nối đầy đủ (Fully Connected). Cuối cùng, hàm kích hoạt Softmax tính toán phân phối xác suất để đưa ra dự đoán chính xác nhất là 'Zebra' (Ngựa vằn) với tỷ lệ 0.7.

2.2.3 Mạng nơ-ron hồi tiếp

Mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) là một lớp mô hình học sâu được thiết kế chuyên biệt cho việc xử lý dữ liệu tuần tự và dữ liệu phụ thuộc theo thời gian. Không giống như các mạng nơ-ron truyền thống giả định các mẫu dữ liệu là độc lập, RNN cho phép mô hình học được mối quan hệ giữa các phần tử trong một chuỗi bằng cách duy trì trạng thái nội tại qua các bước thời gian. Nhờ đặc tính này, RNN được ứng dụng rộng rãi trong các bài toán như dự báo chuỗi thời gian, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói và phân tích hành vi theo thời gian.

Cấu trúc của một mạng RNN điển hình có thể được mô tả thông qua các thành

phần chính sau:

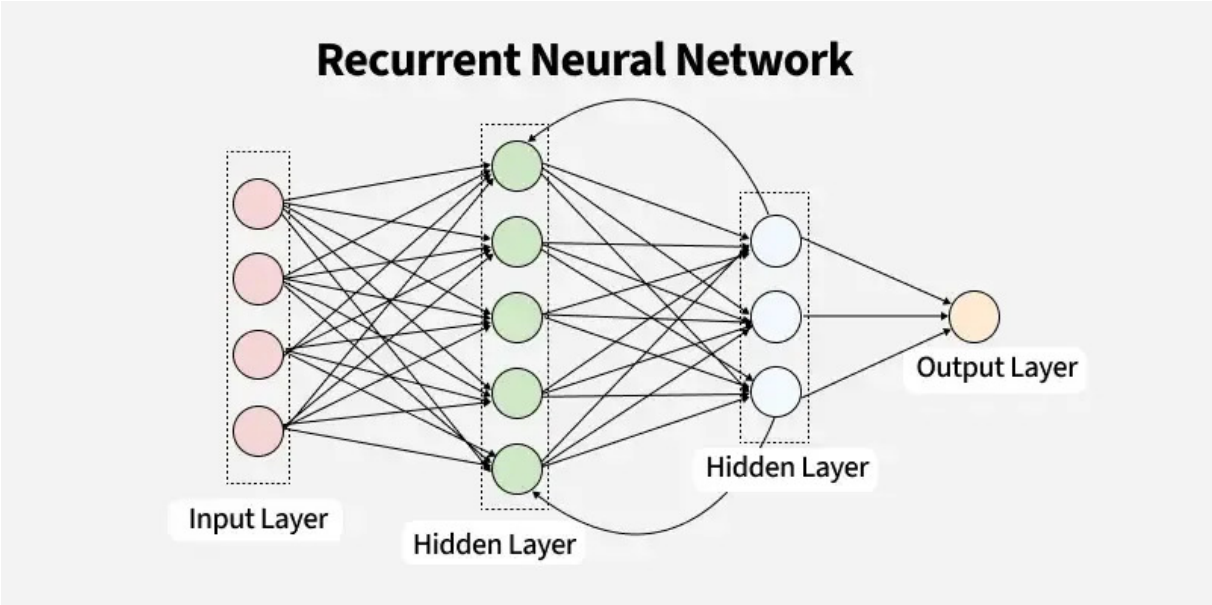
- **Dữ liệu đầu vào theo chuỗi:** Dữ liệu đầu vào của RNN được tổ chức dưới dạng chuỗi các phần tử theo thứ tự thời gian, ký hiệu là $\{x_1, x_2, \ldots, x_T\}$, trong đó mỗi x_t biểu diễn dữ liệu tại thời điểm t. Trong bài toán giao thông, các phần tử này có thể là lưu lượng xe, mật độ phương tiện hoặc các đặc trưng được trích xuất từ ảnh camera theo từng khung thời gian.

- **Trạng thái ẩn (hidden state):** Trạng thái ẩn h_t đóng vai trò là bộ nhớ tạm thời của mạng, lưu giữ thông tin từ các bước thời gian trước. Tại mỗi thời điểm, trạng thái ẩn được cập nhật dựa trên dữ liệu đầu vào hiện tại x_t và trạng thái trước đó h_{t-1}. Cơ chế này cho phép RNN mô hình hóa các mối quan hệ phụ thuộc theo thời gian trong dữ liệu.

- **Trọng số hồi tiếp (Recurrent Weights):** RNN sử dụng các trọng số được chia sẻ qua các bước thời gian, bao gồm trọng số kết nối từ đầu vào đến trạng thái ẩn và trọng số hồi tiếp từ trạng thái ẩn trước sang trạng thái ẩn hiện tại. Việc chia sẻ trọng số giúp mô hình giảm số lượng tham số và duy trì tính nhất quán trong việc xử lý chuỗi dữ liệu.

- **Hàm kích hoạt (Activation Function):** Sau khi thực hiện phép biến đổi tuyến tính, trạng thái ẩn thường được đưa qua các hàm kích hoạt phi tuyến như tanh hoặc ReLU. Các hàm này giúp mạng học được các mối quan hệ phi tuyến trong dữ liệu chuỗi.

- **Đầu ra (Output):** Tại mỗi bước thời gian hoặc tại bước cuối của chuỗi, RNN có thể sinh ra đầu ra y_t tương ứng. Đầu ra này có thể được sử dụng cho nhiều mục đích khác nhau như phân loại, hồi quy hoặc dự báo giá trị trong tương lai.



Hình 2.2.4: Sơ đồ cấu trúc mạng nơ-ron hồi tiếp (RNN). Hình ảnh minh họa các lớp đầu vào (Input), hai lớp ẩn (Hidden Layers) và lớp đầu ra (Output), với đặc điểm nhận dạng chính là các mũi tên cong (feedback loops) kết nối ngược từ lớp ẩn sau về lớp ẩn trước.

2.3 TỔNG QUAN VỀ MÔ HÌNH LSTM

2.3.1 Động lực ra đời của LSTM

Mạng nơ-ron hồi tiếp (Recurrent Neural Network - RNN) được thiết kế nhằm xử lý các dạng dữ liệu có tính tuần tự, trong đó thông tin tại thời điểm hiện tại phụ thuộc vào các trạng thái trước đó. Tuy nhiên, trong thực tế, RNN truyền thống gặp phải hạn chế nghiêm trọng khi học các phụ thuộc dài hạn trong chuỗi thời gian, đặc biệt là hiện tượng tiêu biến gradient (vanishing gradient) và bùng nổ gradient (exploding gradient) trong quá trình lan truyền ngược theo thời gian (Backpropagation Through Time). Hệ quả là mô hình khó có thể ghi nhớ và khai thác hiệu quả các thông tin xảy ra ở những thời điểm xa trong quá khứ, làm suy giảm độ chính xác trong các bài toán dự báo dài hạn.

Trong nhiều bài toán thực tiễn, đặc biệt là dự báo chuỗi thời gian, mối quan hệ giữa dữ liệu không chỉ tồn tại ở các bước thời gian gần nhau mà còn kéo dài trong khoảng thời gian lớn. Ví dụ, trong lĩnh vực giao thông đô thị, lưu lượng xe tại một thời điểm chịu ảnh hưởng không chỉ bởi vài phút trước đó mà còn bởi các yếu tố mang tính chu kỳ như giờ cao điểm, ngày trong tuần hoặc các sự kiện định kỳ. RNN truyền thống khó nắm bắt

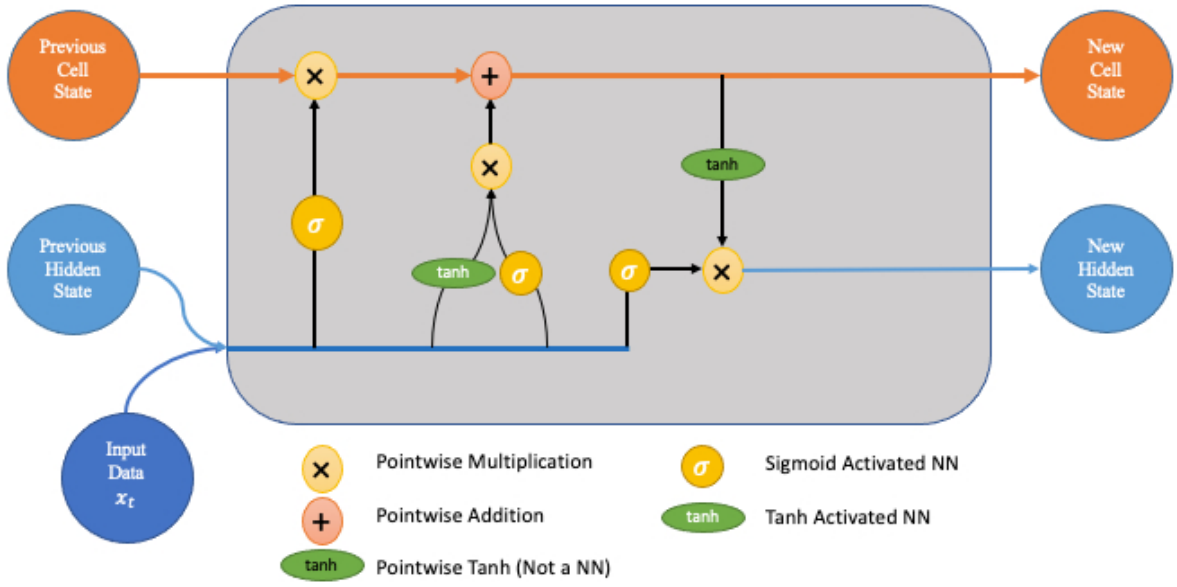
được những mối quan hệ dài hạn này, dẫn đến khả năng dự báo kém ổn định và thiếu chính xác.

Xuất phát từ những hạn chế trên, mạng Long Short-Term Memory (LSTM) được đề xuất nhằm cải thiện khả năng ghi nhớ thông tin dài hạn của RNN. Bằng cách đưa vào một cơ chế bộ nhớ riêng (Cell State) cùng với các cổng điều khiển (Forget Gate, Input Gate và Output Gate), LSTM cho phép mô hình chủ động quyết định thông tin nào cần được lưu giữ, cập nhật hoặc loại bỏ theo thời gian. Nhờ đó, LSTM khắc phục hiệu quả vấn đề tiêu biến gradient, đồng thời nâng cao khả năng mô hình hóa các phụ thuộc dài hạn trong chuỗi dữ liệu. Đây chính là lý do LSTM trở thành một trong những kiến trúc phổ biến và hiệu quả nhất cho các bài toán dự báo chuỗi thời gian, bao gồm dự báo lưu lượng và trạng thái giao thông đô thị.

2.3.2 Kiến trúc cơ bản của LSTM

Thành phần cốt lõi của một khối LSTM là bộ nhớ trạng thái (Cell State - c_t), hoạt động như một băng chuyền thông tin chạy xuyên suốt chuỗi thời gian với các tương tác tuyến tính tối thiểu, giúp thông tin được bảo toàn nguyên vẹn qua nhiều bước xử lý. Ngoài ra, còn có lớp ẩn trước đó (previous hidden state - h_{t-1}) đón vai trò như là đầu ra của 1 điểm trước đó và dữ liệu đầu vào tại thời điểm hiện tại (x_t).

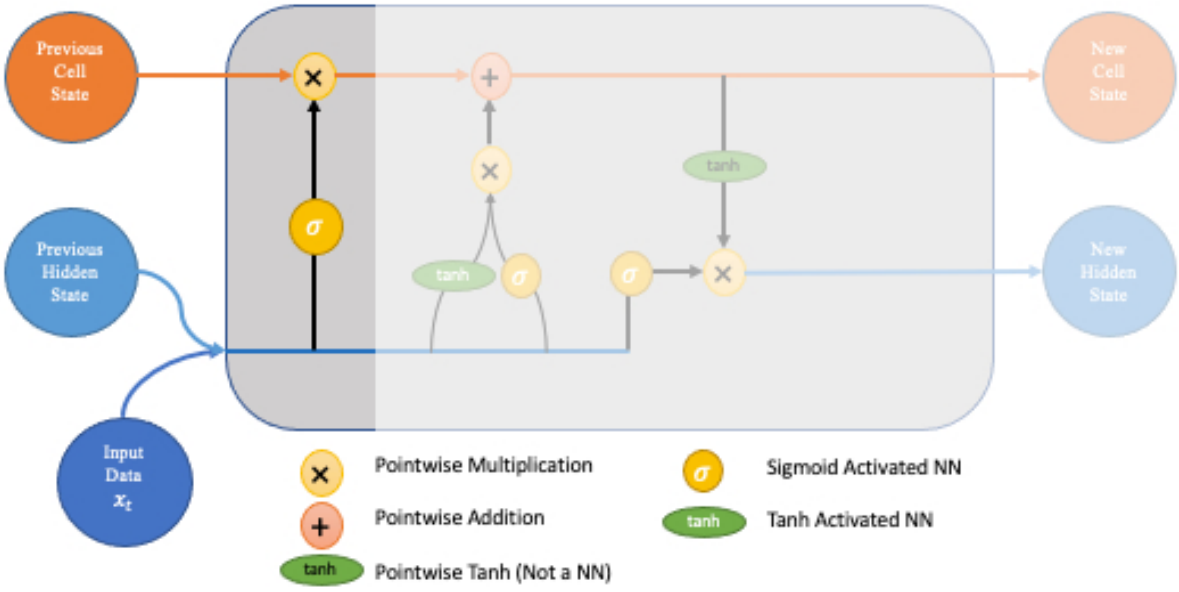
Bên cạnh ba thành phần quan trọng trên, LSTM còn sỡ hữu ba cổng điều khiển (gates) chính: Quên (Forget Gate), Đầu vào (Input gate) và Đầu ra (Output gate). Ba cổng này cùng với bộ nhớ trạng thái, các lớp ẩn và dữ liệu đầu vào hợp nên kiến trúc cơ bản của một khối LSTM như hình **??** minh họa. [**?**]



Hình 2.3.5: Sơ đồ kiến trúc mạng LSTM

2.3.3 Cơ chế hoạt động của các cổng trong LSTM

Cổng quên (Forget Gate)



Hình 2.3.6: Sơ đồ cổng quên trong LSTM (Phần được in đậm)

Bước đầu tiên trong quá trình xử lý của LSTM là quyết định thông tin nào từ Cell State trước đó (c_{t-1}) cần được giữ lại hoặc loại bỏ. Để làm được điều đó, lớp ẩn trước

đó (h_{t-1}) và dữ liệu đầu vào hiện tại (x_t) được đưa vào một mạng nơ-ron với hàm kích hoạt sigmoid (σ). Hàm sigmoid sẽ tạo ra một vector có giá trị trong khoảng [0,1], đảm bảo rằng đầu ra của hàm sigmoid sẽ tiến tới 0 khi dữ liệu đầu vào được cho là không liên quan và tiến tới 1 khi dữ liệu liên quan hơn.

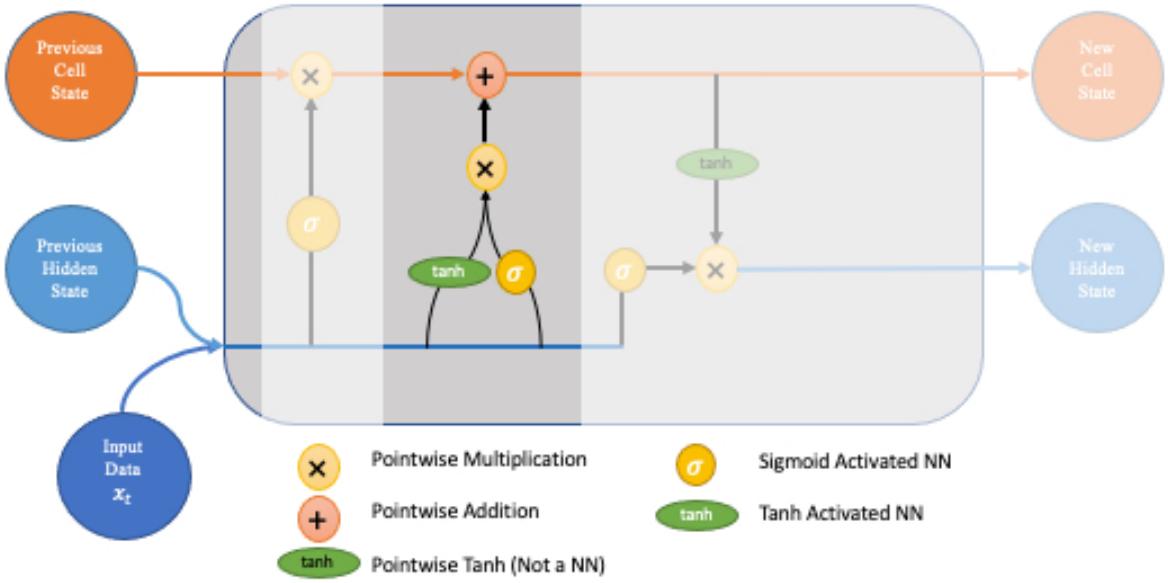
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.3.1}$$

Trong đó:

- f_t là đầu ra của Forget Gate tại thời điểm t.
- W_f là ma trận trọng số của liên kết giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại của Forget Gate.
- b_f là giá trị bias giúp điều chỉnh quyết định của Forget Gate.
- $[h_{t-1}, x_t]$ là phép nối (concatenate) giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại.
- σ là hàm kích hoạt sigmoid.

Đầu ra f_t của Forget Gate sẽ được sử dụng để nhân với Cell State trước đó (c_{t-1}), quyết định phần thông tin nào sẽ được giữ lại trong bộ nhớ.

Cổng đầu vào (Input Gate)



Hình 2.3.7: Sơ đồ cổng đầu vào trong LSTM (Phần được in đậm)

Input Gate chịu trách nhiệm quyết định thông tin mới nào sẽ được thêm vào Cell State.Có thể nhìn thấy theo hình **??**, quá trình cập nhật thông tin mới boa gồm sự phối hợp giữa Input Gate và vector ứng viên mới. Cụ thể:

- Một vector ứng viên mới được tổng hợp từ lớp ẩn trước đó (h_{t-1}) và dữ liệu đầu vào hiện tại (x_t) thông qua một mạng nơ-ron với hàm kích hoạt tanh. Vector này biểu diễn các giá trị tiềm năng để cập nhật vào Cell State. Bởi vì giá trị hàm tanh trong khoảng [-1, 1], nó giúp điều chỉnh các giá trị ứng viên mới sao cho phù hợp với phạm vi của Cell State, tránh việc giá trị trở nên quá lớn.
- Một mạng nơ-ron khác với hàm kích hoạt sigmoid được sử dụng như một bộ lọc để xác định mức độ quan trọng của từng giá trị trong vector ứng viên mới. Đầu ra của mạng này sẽ quyết định phần thông tin nào trong vector ứng viên sẽ được thêm vào Cell State.

Output của Input Gate được tính toán như sau:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.3.2}$$

Trong đó:

- i_t là đầu ra của Input Gate tại thời điểm t.
- W_i là ma trận trọng số của liên kết giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại của Input Gate.
- b_i là giá trị bias giúp điều chỉnh quyết định của Input Gate.
- $[h_{t-1}, x_t]$ là phép nối (concatenate) giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại.
- σ là hàm kích hoạt sigmoid.

Vector ứng viên mới được tính toán như sau:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.3.3}$$

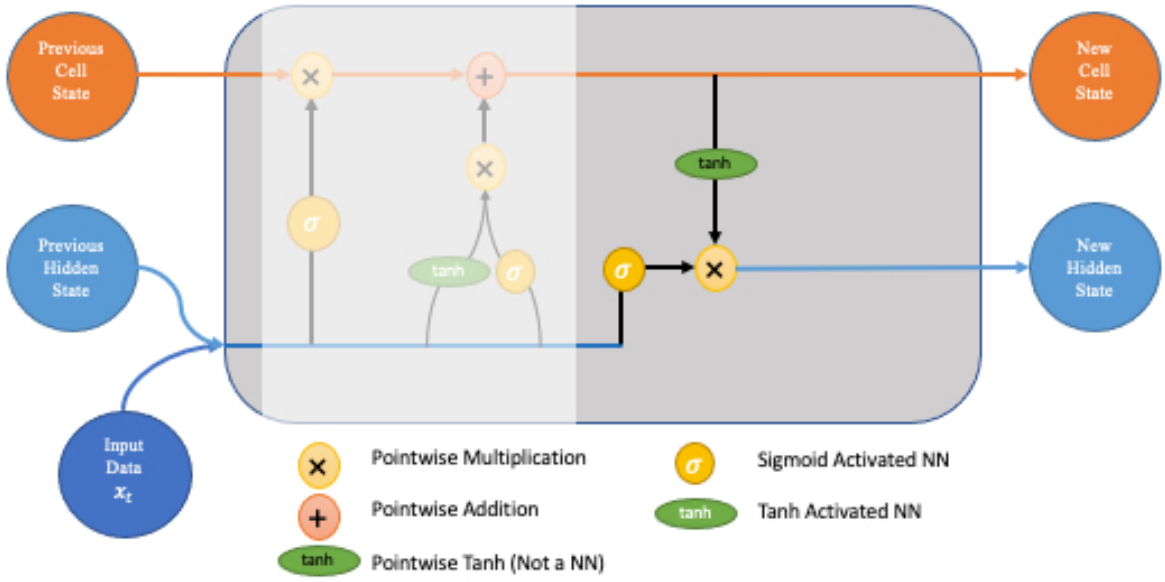
Trong đó:

- $\tilde{c}_t$ là vector ứng viên mới tại thời điểm t.
- W_c là ma trận trọng số của liên kết giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại để tạo ra vector ứng viên mới.
- b_c là giá trị bias giúp điều chỉnh quá trình tạo ra vector ứng viên mới.
- $[h_{t-1}, x_t]$ là phép nối (concatenate) giữa lớp ẩn trước đó và dữ liệu đầu vào hiện tại.
- tanh là hàm kích hoạt tanh.

Đầu ra của 2 bước trên sẽ được sử dụng để cập nhật Cell State hiện tại (c_t) như sau:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{2.3.4}$$

Cổng đầu ra (Output Gate)



Hình 2.3.8: Sơ đồ cổng đầu ra trong LSTM (Phần được in đậm)

Việc cập nhật dữ liệu cho Cell State đã hoàn tất, bước tiếp theo cũng là bước cuối cùng trong quá trình xử lý của LSTM là quyết định một lớp ẩn mới. Để làm điều này, LSTM sử dụng 3 thứ: Cell State mới, lớp ẩn trước đó và dữ liệu đầu vào hiện tại. Lý do cho việc tại sao lại không đưa cell state mới ra làm đầu ra cho Output Gate (Tức là lớp ẩn tiếp theo) là bởi vì Cell State chứa toàn bộ thông tin theo thời gian và không phải tất cả thông tin đều cần thiết cho việc dự báo tại thời điểm hiện tại. Do đó, Output Gate sẽ quyết định phần thông tin nào từ Cell State mới sẽ được sử dụng để tạo ra lớp ẩn mới.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) h_t = o_t \cdot \tanh(c_t) \tag{2.3.5}$$

Tuy nhiên, lớp ẩn mới (h_t) là được coi là output của mô hình LSTM nhưng để ra được kết quả cuối cùng mà con người hiểu được, ta cần thêm một bước nữa là đưa lớp ẩn mới qua một lớp Linear Layer (Dense Layer).

2.3.4 Ứng dụng của LSTM trong dự báo chuỗi thời gian

Mạng bộ nhớ dài-ngắn hạn (LSTM) đại diện cho một bước tiến quan trọng trong lĩnh vực học sâu, đặc biệt là trong việc giải quyết các bài toán dự báo chuỗi thời gian phức tạp nhờ vào khả năng xử lý các phụ thuộc dài hạn. Khác với các mô hình mạng

thần kinh tái phát truyền thống thường gặp phải hiện tượng triệt tiêu đạo hàm khi huấn luyện trên các chuỗi dữ liệu dài, LSTM được thiết kế với cấu trúc tế bào đặc biệt bao gồm các cổng kiểm soát luồng thông tin một cách tinh vi. Nhờ vào sự phối hợp giữa cổng quên, cổng vào và cổng ra, mô hình có khả năng chọn lọc thông tin quan trọng để lưu giữ trong trạng thái tế bào qua nhiều bước thời gian và loại bỏ những dữ liệu nhiễu không còn giá trị. Đặc tính này giúp LSTM vượt trội trong việc nắm bắt các quy luật lặp lại, xu hướng và các mối liên hệ nhân quả tiềm ẩn trong dữ liệu lịch sử. Trong lĩnh vực tài chính và kinh tế, LSTM đã trở thành một công cụ chủ đạo để dự báo các biến số có độ biến động cao như giá cổ phiếu, tỷ giá hối đoái và chỉ số lạm phát. Do dữ liệu tài chính thường chứa đựng nhiều yếu tố phi tuyến tính và chịu ảnh hưởng bởi tâm lý thị trường phức tạp, các phương pháp thống kê cổ điển như ARIMA thường khó có thể phản ánh đầy đủ các biến chuyển bất ngờ. LSTM có khả năng tự động trích xuất các đặc trưng từ dữ liệu quá khứ mà không cần các giả định khắt khe về phân phối xác suất, từ đó giúp các nhà đầu tư và hoạch định chính sách nhận diện được xu hướng tăng trưởng hoặc rủi ro suy thoái với độ chính xác cao hơn. Việc tích hợp LSTM vào các hệ thống giao dịch tự động cũng cho phép tối ưu hóa danh mục đầu tư dựa trên việc dự đoán các điểm đảo chiều của thị trường trong ngắn hạn và dài hạn. Đối với ngành năng lượng và quản lý hạ tầng, ứng dụng của LSTM tập trung mạnh mẽ vào việc dự báo nhu cầu tiêu thụ điện năng và quản lý lưới điện thông minh. Việc dự báo chính xác tải trọng điện đóng vai trò quyết định trong việc cân bằng cung cầu, giảm thiểu lãng phí và lập kế hoạch vận hành cho các nguồn năng lượng tái tạo như điện gió hay điện mặt trời vốn có tính bất định cao. LSTM chứng minh được hiệu quả vượt trội khi xử lý các chuỗi dữ liệu có tính chu kỳ đa dạng, từ chu kỳ theo giờ trong ngày đến chu kỳ theo mùa trong năm, đồng thời tích hợp hiệu quả các biến số ngoại sinh như điều kiện thời tiết và các sự kiện kinh tế xã hội. Tương tự, trong lĩnh vực giao thông vận tải, mô hình này được sử dụng để dự đoán lưu lượng xe cộ và thời gian di chuyển, hỗ trợ việc điều tiết giao thông và giảm thiểu ùn tắc tại các đô thị lớn. Lĩnh vực y tế và khoa học đời sống cũng chứng kiến những đóng góp quan trọng của LSTM trong việc giám sát sức khỏe bệnh nhân và dự báo xu hướng dịch bệnh. Thông qua việc phân tích các tín hiệu sinh học liên tục như điện tâm đồ, điện não đồ hoặc các chỉ số sinh tồn được thu thập từ thiết bị đeo thông minh, LSTM có khả năng phát hiện sớm các dấu hiệu bất thường và đưa ra cảnh báo về các biến cố lâm sàng trước khi chúng xảy ra. Trong công tác y tế cộng đồng, mô hình này hỗ trợ dự đoán sự

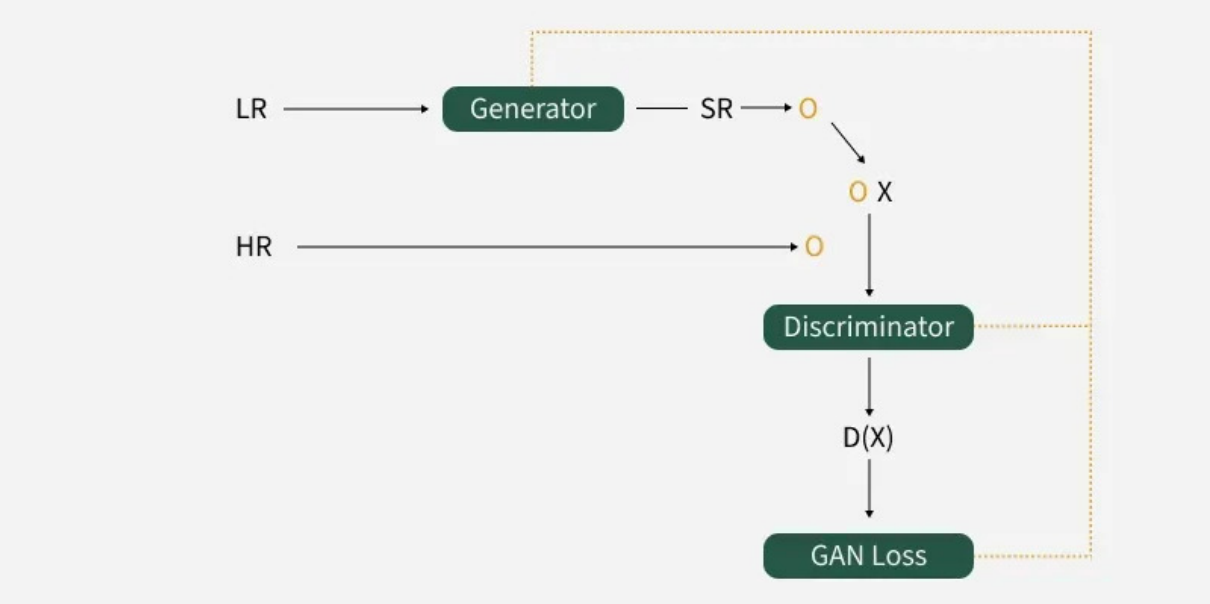
lây lan của các bệnh truyền nhiễm dựa trên dữ liệu lịch sử về ca nhiễm, biến động thời tiết và mật độ dân cư. Khả năng xử lý các dữ liệu chuỗi thời gian không đồng nhất giúp các cơ quan y tế có cái nhìn toàn diện để đưa ra các biện pháp can thiệp kịp thời và hiệu quả. Mặc dù sở hữu những ưu điểm vượt trội về khả năng ghi nhớ, việc triển khai LSTM trong thực tế vẫn đòi hỏi sự cân nhắc kỹ lưỡng về chi phí tính toán và yêu cầu về khối lượng dữ liệu huấn luyện. Quá trình huấn luyện một mô hình LSTM thường yêu cầu tài nguyên phần cứng lớn và kỹ năng tinh chỉnh các siêu tham số để tránh hiện tượng quá khớp, đặc biệt khi làm việc với các chuỗi thời gian có nhiều nhiễu. Tuy nhiên, với sự phát triển của các kiến trúc lai phối hợp giữa LSTM với các cơ chế chú ý hoặc mạng thần kinh tích chập, khả năng biểu diễn của mô hình ngày càng được nâng cao. Những cải tiến này không chỉ giúp tăng tốc độ hội tụ mà còn cho phép mô hình tập trung vào những phân đoạn thời gian quan trọng nhất, khẳng định vị thế của LSTM như một trụ cột trong hệ thống các phương pháp dự báo hiện đại.

2.4 TỔNG QUAN VỀ SUPER RESOLUTION GENERATIVE ADVERSARIAL NETWORK

Super-Resolution Generative Adversarial Network (SRGAN), được Ledig và cộng sự giới thiệu vào năm 2016, là một trong những mô hình tiên phong ứng dụng mạng GAN vào bài toán tăng độ phân giải ảnh. Mục tiêu của SRGAN là khắc phục hạn chế của các phương pháp nội suy truyền thống và các mô hình tối ưu theo lỗi điểm ảnh như MSE, vốn thường tạo ra ảnh mượt nhưng thiếu chi tiết và không giữ được kết cấu tự nhiên. Thông qua cơ chế huấn luyện đối kháng, SRGAN học cách sinh ra ảnh có độ phân giải cao với chi tiết sắc nét hơn bằng cách sử dụng đồng thời hai thành phần: perceptual loss dựa trên đặc trưng trích xuất từ mạng VGG và adversarial loss từ Discriminator. Sự kết hợp này giúp mô hình tái tạo các hoa văn, kết cấu và đường nét tinh vi thường bị mất đi trong quá trình phóng to ảnh, từ đó tạo ra ảnh đầu ra có chất lượng thị giác chân thực và giàu chi tiết hơn so với các kỹ thuật SR truyền thống. [**?**]

2.4.1 Tổng quan kiến trúc

SRGAN hoạt động theo cơ chế GAN truyền thống, trong đó có hai mạng nơ-ron đóng vai trò đối kháng nhau: Generator nhận ảnh độ phân giải thấp và sinh ra phiên bản độ phân giải cao, trong khi Discriminator cố gắng phân biệt ảnh thật với ảnh được tạo ra. Quá trình huấn luyện xen kẽ này buộc Generator phải liên tục cải thiện chất lượng ảnh sinh ra để ngày càng giống với ảnh thật hơn. [?]



Hình 2.4.9: Kiến trúc SRGAN

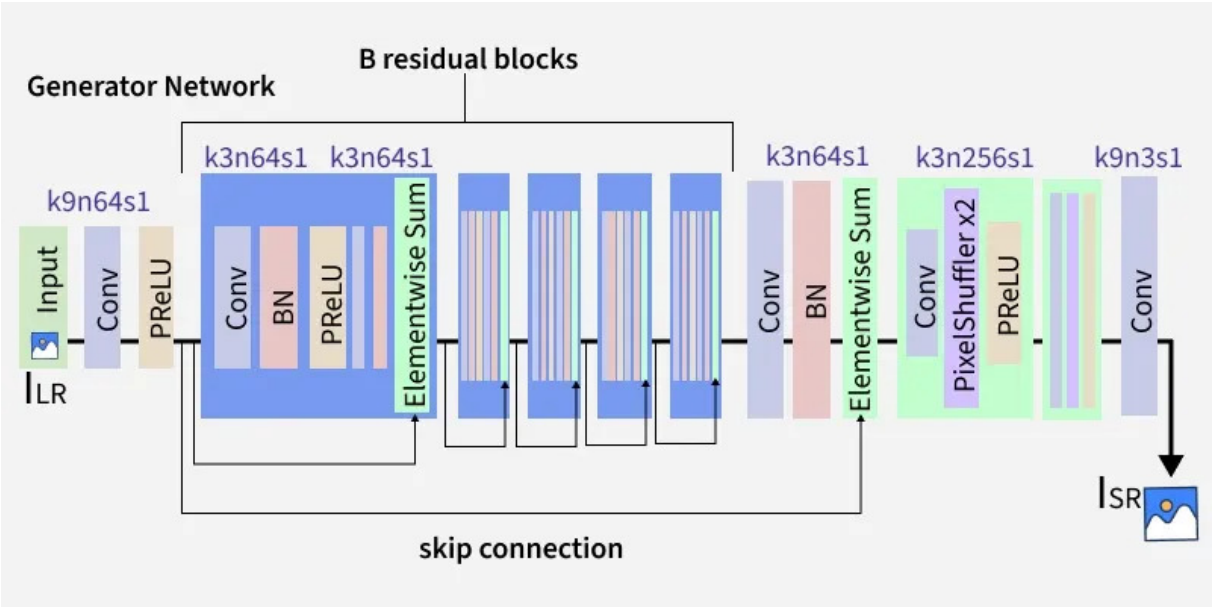
2.4.2 Kiến trúc Generator

Generator sử dụng kiến trúc mạng Residual Network (ResNet) thay vì các mạng tích chập sâu thông thường. Việc lựa chọn ResNet đóng vai trò quan trọng bởi các kết nối tắt (skip connections) trong kiến trúc này giúp dòng gradient lan truyền hiệu quả hơn trong quá trình huấn luyện. Nhờ đó, mô hình có thể xây dựng các mạng rất sâu mà không gặp phải hiện tượng mất mát gradient, đồng thời cải thiện khả năng học đặc trưng tinh vi trong ảnh.

Bộ sinh (Generator) được xây dựng từ 16 Residual Block, mỗi khối gồm hai lớp tích chập với kernel kích thước 3x3 và 64 kênh đặc trưng. Sau mỗi lớp tích chập là Batch Normalization và hàm kích hoạt Parametric ReLU (PReLU). Khác với ReLU hoặc

LeakyReLU truyền thống, PReLU cho phép hệ số dốc ở vùng âm được học tự động, giúp mô hình thích nghi tốt hơn trong quá trình huấn luyện mà không làm tăng nhiều chi phí tính toán.

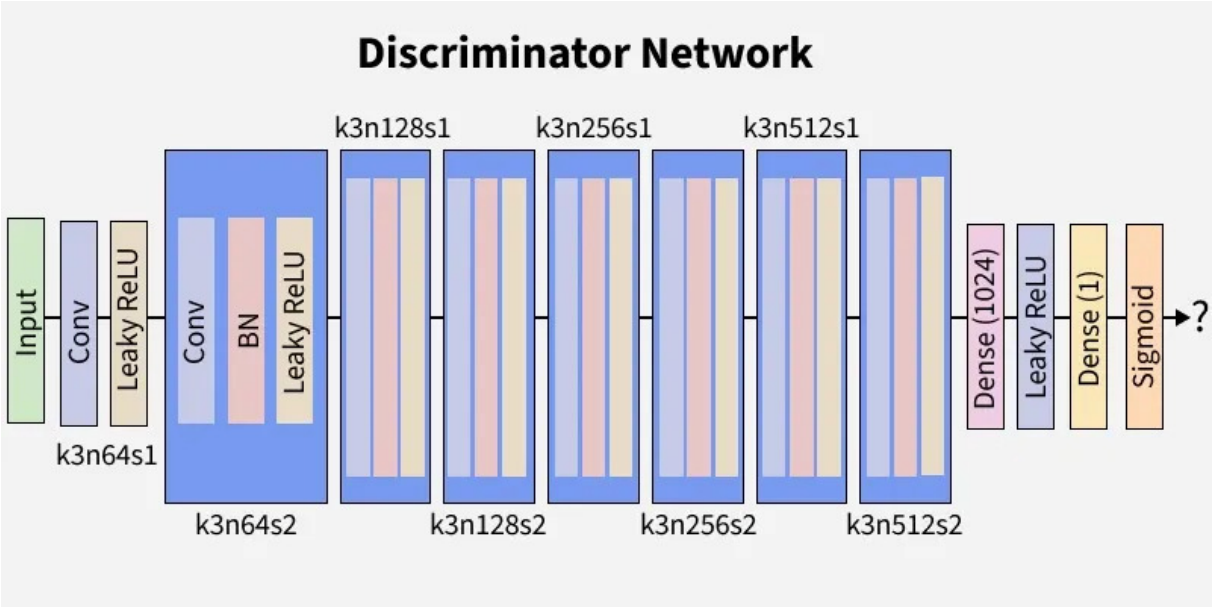
Giai đoạn tăng độ phân giải của mô hình được thực hiện thông qua hai lớp sub-pixel convolution đã được huấn luyện, giúp phóng to kích thước không gian một cách hiệu quả. Phương pháp này hoạt động bằng cách tái sắp xếp thông tin từ chiều kênh sang chiều không gian, cho phép mô hình học trực tiếp cách upsample ảnh thay vì sử dụng các kỹ thuật nội suy đơn giản.



Hình 2.4.10: Kiến trúc của bộ Generator

2.4.3 Kiến trúc Discriminator

Bộ phân biệt (Discriminator) được thiết kế theo kiến trúc sâu gồm tám lớp tích chập 3x3, trong đó số lượng đặc trưng được mở rộng dần từ 64 lên 512 khi kích thước không gian giảm xuống thông qua các lớp tích chập có bước nhảy (strided convolution). Sau khối trích xuất đặc trưng này, mô hình sử dụng hai lớp kết nối đầy đủ (fully connected) và kết thúc bằng hàm kích hoạt sigmoid, nhằm đưa ra xác suất thể hiện mức độ "thật" của ảnh đầu vào—tức phân biệt ảnh thực với ảnh do bộ sinh tạo ra.



Hình 2.4.11: Kiến trúc của bộ Discriminator

2.4.4 Thiết kế hàm mất mát

Content Loss

Trong các phương pháp super-resolution truyền thống, Mean Squared Error (MSE) thường được sử dụng làm hàm mất mát, đo lường sai khác điểm ảnh giữa ảnh sinh và ảnh đích. Tuy nhiên, việc tối ưu theo MSE thường khiến ảnh tái tạo trở nên quá trơn mượt, thiếu chi tiết. Nguyên nhân là MSE thúc đẩy mô hình tạo ra một kết quả "trung bình" trong số nhiều khả năng khôi phục ảnh độ phân giải cao tương ứng với một ảnh đầu vào bị giảm chất lượng, từ đó làm mất đi các cấu trúc tinh tế và độ sắc nét vốn có.

$$l^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y}\right)^2 \tag{2.4.6}$$

Trong đó:

- $l^{SR}_{VGG/i,j}$: Giá trị hàm mất mát (VGG) tại lớp (i, j).
- $W_{i,j}, H_{i,j}$: Chiều rộng và chiều cao của bản đồ đặc trưng VGG tại tầng (i, j), dùng để chuẩn hóa.
- $\phi_{i,j}$: Bản đồ đặc trưng được trích xuất từ lớp (i, j) của mạng VGG đã được huấn

luyện trước.

- I^{HR}: Ảnh độ phân giải cao thực tế (ground-truth).
- I^{LR}: Ảnh đầu vào có độ phân giải thấp.
- $G_{\theta_G}(I^{LR})$: Ảnh độ phân giải cao được sinh ra bởi bộ Generator G.
- (x, y): Vị trí không gian một điểm trong bản đồ đặc trưng.

Adversarial Loss

Adversarial loss đóng vai trò thúc đẩy bộ sinh tạo ra các ảnh mà bộ phân biệt không thể phân tách với ảnh độ phân giải cao thực sự. Thành phần mất mát này đặc biệt quan trọng trong việc khôi phục các chi tiết sắc nét và kết cấu chân thực, giúp ảnh được phóng đại trở nên tự nhiên và thuyết phục hơn về mặt thị giác.

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{2.4.7}$$

Trong đó:

- l_{Gen}^{SR}: Giá trị Adversarial loss cho bộ Generator.
- N: Số lượng mẫu ảnh.
- $D_{\theta_D}(G_{\theta_G}(I^{LR}))$: Xác suất mà bộ Discriminator đánh giá ảnh được sinh ra là ảnh thật.
- $G_{\theta_G}(I^{LR})$: Ảnh độ phân giải cao được sinh ra bởi bộ Generator sử dụng ảnh đầu vào có độ phân giải thấp I^{LR}.
- $-logD_{\theta D}(G_{\theta G}(I^{LR}))$: Phạt Generator khi Discriminator dễ dàng nhận ra ảnh giả mà nó tạo ra.

Tổng hợp hàm mất mát

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR} \tag{2.4.8}$$

Trong đó:

- l^{SR}: Hàm mất mát tổng hợp cho super-resolution.
- l_X^{SR}: Content loss dựa trên đặc trưng VGG perceptual.
- l_{Gen}^{SR}: Adversarial loss từ bộ Generator.
- 10^{-3}: Hệ số trọng số cân bằng giữa hai thành phần mất mát, đảm bảo Content Loss chiếm ưu thế trong quá trình huấn luyện.

2.5 TỔNG QUAN VỀ ESRGAN VÀ REAL-ESRGAN

2.5.1 ESRGAN

ESRGAN (Enhanced Super-Resolution GAN) là phiên bản cải tiến của SRGAN, tiếp tục khai thác sức mạnh của mô hình GAN trong bài toán tăng độ phân giải ảnh. ESRGAN khắc phục những hạn chế của SRGAN, đồng thời nâng cao chất lượng thị giác bằng cách tái tạo nhiều chi tiết tinh vi và sắc nét hơn. Những cải tiến đáng chú ý của ESRGAN gồm: [**?**]

- **Residual-in-Residual Dense Block (RRDB):** Cấu trúc khối mới giúp tăng cường khả năng học đặc trưng và cải thiện độ ổn định khi huấn luyện, thay thế cho các residual block truyền thống.
- **Nâng cấp perceptual loss:** Giúp mô hình tạo ra hình ảnh tự nhiên và chân thực hơn.
- **Relativistic GAN:** ESRGAN sử dụng hàm mất mát GAN mang tính tương đối (Relativistic GAN loss) thay cho GAN cổ điển, nhằm giúp bộ phân biệt đánh giá ảnh thật có "tính chân thực cao hơn tương đối" so với ảnh giả, thay vì chỉ đánh giá theo dạng nhị phân thật/giả.

Các thành phần chính trong cấu trúc:

- **Generator:** Nhiệm vụ của bộ sinh là chuyển đổi ảnh đầu vào độ phân giải thấp (LR) thành ảnh độ phân giải cao (HR). Trong kiến trúc ESRGAN, các khối RRDB đóng vai trò trung tâm trong việc trích xuất đặc trưng và phóng đại ảnh, đây cũng là điểm cải tiến quan trọng so với SRGAN.

- **Discriminator:** Bộ phân biệt được huấn luyện để phân loại xem một ảnh là ảnh HR thật hay ảnh SR do generator tạo ra. Mục tiêu của nó là phát hiện chính xác các ảnh giả nhằm thúc đẩy generator tạo ra hình ảnh ngày càng chân thực.

- **Perceptual Loss:** Một cải tiến quan trọng của ESRGAN là sử dụng hàm mất mát cảm nhận, đo mức độ tương đồng thị giác giữa ảnh sinh và ảnh thật bằng cách so sánh các bản đồ đặc trưng trích xuất từ mạng VGG đã được huấn luyện trước. Điều này giúp ảnh SR có tính tự nhiên và dễ chịu hơn đối với người quan sát.

Các hàm mất mát trong ESRGAN:

- **Content Loss:** Đo sự khác biệt giữa ảnh HR thật và ảnh HR được mô hình tạo ra, thường tính theo mức pixel với chỉ số MSE.

- **Adversarial Loss:** Bảo đảm rằng ảnh sinh trông càng giống ảnh thật càng tốt bằng việc tối ưu dựa trên phản hồi từ discriminator.

- **Perceptual Loss:** So sánh các đặc trưng bậc cao giữa ảnh sinh và ảnh thật, nhằm duy trì chất lượng thị giác, chi tiết và kết cấu trong ảnh.

2.5.2 Real-ESRGAN

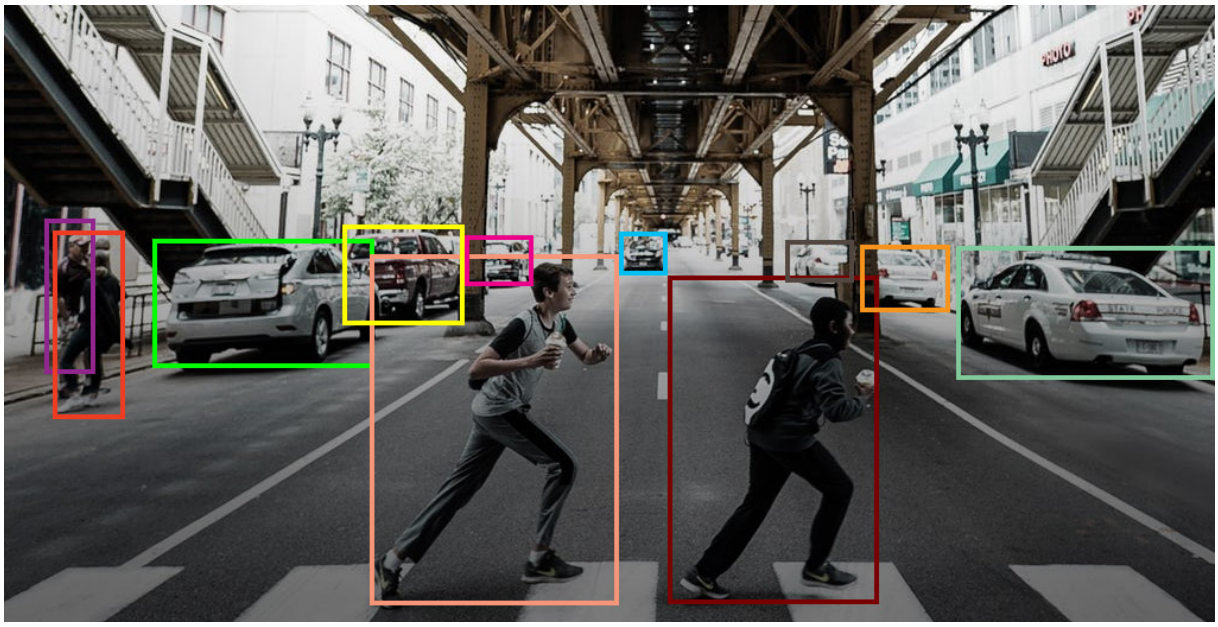
Real-ESRGAN (Real-World Enhanced Super-Resolution Generative Adversarial Network) được phát triển nhằm khắc phục các hạn chế của ESRGAN khi xử lý dữ liệu ngoài đời thực. Trong khi ESRGAN chủ yếu hoạt động tốt trên các tập dữ liệu tổng hợp và giả lập—nơi nhiễu, mờ và suy giảm được mô hình hóa đơn giản—Real-ESRGAN hướng tới việc tái tạo ảnh độ phân giải cao từ các đầu vào bị suy giảm phức tạp và không dự đoán được trong thực tế. Mô hình này sử dụng cơ chế GAN bất đối xứng (asymmetric GAN) với bộ suy giảm mạnh mẽ (degradation model) gồm nhiều giai đoạn, mô phỏng các hiện

tượng suy giảm thường gặp như nhiễu cảm biến, nén JPEG, mất chi tiết do chuyển động và sai lệch quang học.

Real-ESRGAN áp dụng kiến trúc RRDB (Residual-in-Residual Dense Block) tương tự ESRGAN nhưng được điều chỉnh để tăng tính ổn định và độ bền khi huấn luyện trên dữ liệu phi chuẩn hóa. Ngoài ra, Real-ESRGAN giới thiệu bộ phân biệt cải tiến (U-shaped discriminator) giúp mô hình học được nhiều mức độ suy giảm khác nhau. Nhờ các cải tiến này, Real-ESRGAN có khả năng tái tạo chi tiết tốt hơn, ổn định hơn trên ảnh chụp thực tế, và cung cấp kết quả nhất quán ngay cả trong điều kiện ảnh đầu vào bị hỏng nghiêm trọng.

2.6 TỔNG QUAN VỀ NHẬN DIỆN ĐỐI TƯỢNG

Một trong những lĩnh vực trọng tâm của Trí tuệ nhân tạo (Artificial Intelligence) là Thị giác máy tính (Computer Vision). Đây là ngành nghiên cứu các phương pháp thu nhận, xử lý và phân tích ảnh số nhằm mô phỏng khả năng "nhìn" và "hiểu" thế giới trực quan của con người. Computer Vision bao gồm nhiều bài toán quan trọng như phân đoạn ảnh, nhận dạng đối tượng, mô phỏng cảnh, siêu phân giải hình ảnh, tái tạo 3D và nhiều hướng tiếp cận khác. Trong số đó, Object Detection được xem là một trong những bài toán cốt lõi và có tác động lớn nhất nhờ tính ứng dụng rộng rãi trong thực tiễn.



Hình 2.6.12: Nhận diện đối tượng

Một số ứng dụng của nhận diện đối tượng bao gồm:

- Phân tích thể thao: Theo dõi vị trí và chuyển động của cầu thủ trên sân.
- Y tế: Phát hiện bất thường trong hình ảnh y khoa.
- Giao thông thông minh: Nhận diện phương tiện, biển báo giao thông, phát hiện vi phạm.
- Thương mại điện tử: Tìm kiếm sản phẩm theo hình ảnh, kiểm tra hàng hóa tự động.
- Nông nghiệp: Giám sát cây trồng, phát hiện sâu bệnh qua hình ảnh.
- Robot tự hành: Giúp robot nhận diện và tương tác với môi trường xung quanh.

Trong những năm gần đây, hai kiến trúc phát hiện đối tượng có ảnh hưởng sâu rộng và hình thành nền tảng cho nhiều hệ thống thị giác máy tính hiện đại là Mạng Nơ-ron Tích chập (Convolutional Neural Networks - CNN) và You Only Look Once (YOLO).

Mạng Nơ-ron Tích chập (CNN) sử dụng phép tích chập như một cơ chế cốt lõi để trích xuất và học các đặc trưng không gian của hình ảnh. Bằng cách áp dụng các bộ

lọc trượt trên toàn bộ ảnh, CNN có khả năng phát hiện các đặc trưng từ thấp đến cao, từ cạnh, góc, cho đến các cấu trúc phức tạp. Từ kiến trúc nền tảng này, nhiều biến thể đã được phát triển nhằm cải thiện tốc độ và độ chính xác, tiêu biểu như R-CNN, Fast R-CNN hay Mask R-CNN, góp phần nâng cao hiệu quả trong các nhiệm vụ định vị và phân loại đối tượng.

YOLO (You Only Look Once) là một trong những mô hình phát hiện đối tượng thời gian thực nổi bật nhất. Được đề xuất lần đầu vào năm 2015, YOLO mang tính đột phá khi tiếp cận bài toán phát hiện đối tượng theo cách hoàn toàn mới: thay vì xử lý theo từng vùng như các mô hình truyền thống, YOLO dự đoán trực tiếp bounding boxes và nhãn lớp chỉ qua một lần quan sát toàn bộ ảnh. Nhờ đó, YOLO đạt được tốc độ xử lý rất cao trong khi vẫn đảm bảo độ chính xác đáng kể, và nhanh chóng trở thành lựa chọn phổ biến trong nhiều ứng dụng thực tế.

Tại Việt Nam, các nghiên cứu đã khai thác mô hình YOLO trong các bài toán như nhận dạng biển số xe và giám sát giao thông, cho thấy hiệu quả trong các hệ thống quản lý và kiểm soát thông minh. Đồng thời, nhiều công trình cũng tiến hành so sánh giữa YOLO và các thuật toán khác như SSD nhằm xác định giải pháp tối ưu cho từng nhu cầu ứng dụng cụ thể.

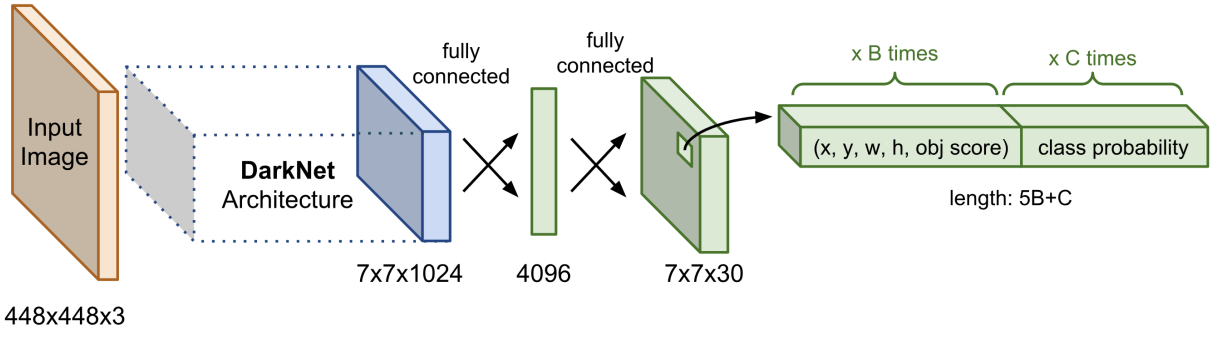
2.7 TỔNG QUAN VỀ MÔ HÌNH YOLO

YOLO (You Only Look Once) là một mô hình phát hiện đối tượng dựa trên mạng nơ-ron tích chập (CNN), được thiết kế để thực hiện nhận dạng, phân loại và định vị đối tượng trong ảnh một cách nhanh chóng và hiệu quả. Không giống như các phương pháp truyền thống chia bài toán thành nhiều giai đoạn (như đề xuất vùng và phân loại), YOLO xử lý toàn bộ ảnh chỉ trong một lần duy nhất, từ đó mang lại tốc độ vượt trội. [**?**] Bản chất "nhìn một lần" của YOLO giúp mô hình đạt tốc độ xử lý thời gian thực mà vẫn duy trì độ chính xác cao, khiến nó trở thành một trong những kiến trúc phổ biến nhất trong lĩnh vực phát hiện đối tượng cho các ứng dụng như giám sát thông minh, xe tự hành và phân tích video. Về mặt kiến trúc, YOLO được cấu trúc thành ba thành phần chính: Backbone, Neck, và Head, mỗi phần đảm nhiệm một vai trò quan trọng trong quá trình

phát hiện đối tượng:

- **Backbone (Mạng trích xuất đặc trưng):** YOLO sử dụng một mạng nơ-ron tích chập sâu (CNN) làm nền tảng để trích xuất đặc trưng từ ảnh đầu vào. Backbone học các thông tin quan trọng như cạnh, hình dạng, họa tiết và cấu trúc đối tượng, tạo nên nền tảng cho quá trình phát hiện chính xác.
- **Neck (Phần kết nối đặc trưng):** Bộ phận này thường tích hợp các kiến trúc như FPN (Feature Pyramid Network) hoặc PANet (Path Aggregation Network). Neck có nhiệm vụ kết hợp và khuếch tán thông tin từ nhiều tầng của Backbone, giúp mô hình duy trì khả năng phát hiện tốt đối với các đối tượng ở nhiều kích thước khác nhau, từ rất nhỏ đến rất lớn.
- **Head (Phần dự đoán đầu ra):** Đây là nơi thực hiện các phép dự đoán cuối cùng, bao gồm tọa độ hộp giới hạn (bounding boxes), điểm tin cậy (confidence scores) và xác suất thuộc lớp đối tượng. Head tổng hợp thông tin từ Backbone và Neck để đưa ra kết quả phát hiện hoàn chỉnh.

Nhờ sự phối hợp hiệu quả giữa ba thành phần này, YOLO đạt được tốc độ xử lý nhanh trong khi vẫn giữ được độ chính xác cao, trở thành một trong những kiến trúc phát hiện đối tượng hiệu quả nhất hiện nay.



Hình 2.7.13: Sơ đồ kiến trúc mạng YOLO

2.7.1 Nguyên lý hoạt động của YOLO

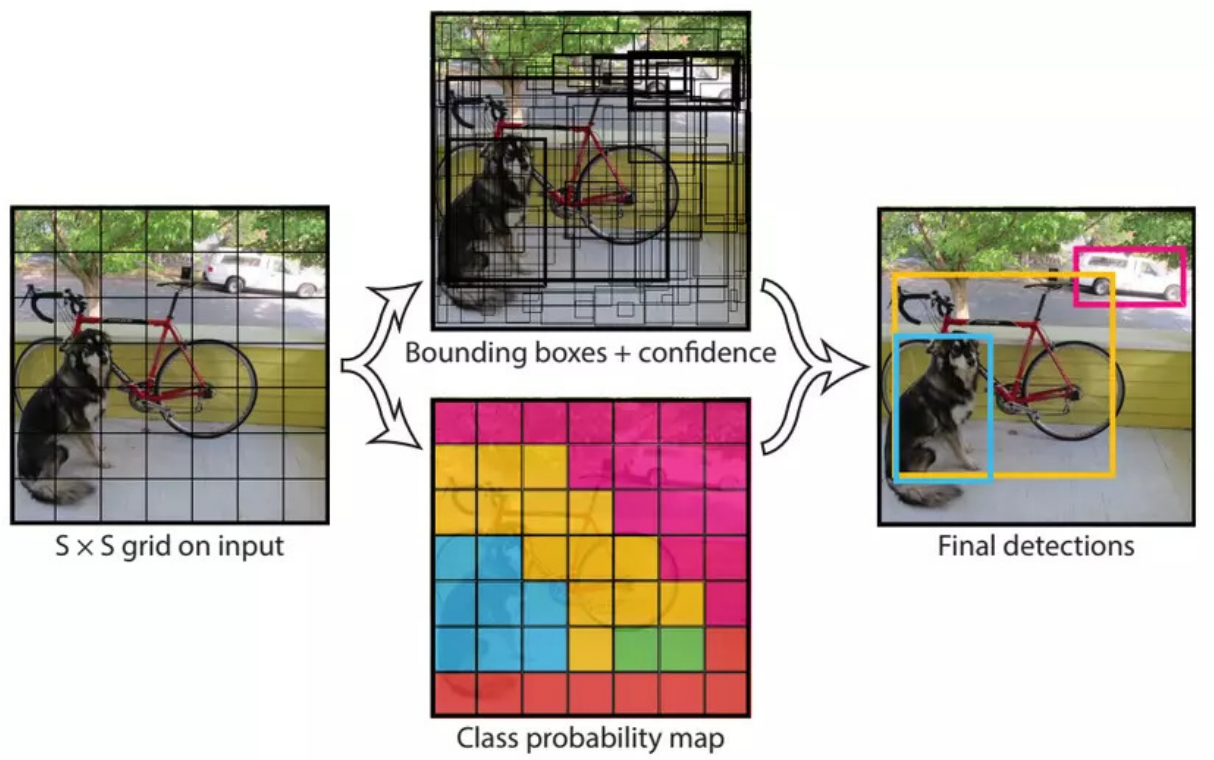
YOLO hoạt động theo 4 bước chính:

- **Chia ảnh thành lưới (grid) S x S:** Ảnh đầu vào được chia thành một lưới gồm S hàng × S cột ô (grid-cells). Mỗi ô "chịu trách nhiệm" phát hiện các đối tượng mà "tâm" của hộp giới hạn (bounding box) rơi vào ô đó. Ý tưởng là phân vùng ảnh để mỗi phần nhỏ có thể dự đoán xem có đối tượng hay không, thay vì lặp toàn bộ ảnh nhiều lần.

- **Trích xuất đặc trưng ảnh với mạng CNN (backbone + feature layers):** YOLO sử dụng một mạng nơ-ron tích chập (CNN) để "đọc" toàn bộ ảnh - các lớp convolution + pooling + ... trích xuất đặc trưng (features) từ ảnh. Các lớp cuối (fully connected / detection layers) sau khi feature extraction sẽ dùng để dự đoán bounding boxes + nhãn + độ tin cậy.

- **Dự đoán nhiều bounding boxes + xác suất + lớp đối tượng ở mỗi ô lưới:** Mỗi ô (grid cell) dự đoán B hộp giới hạn (bounding boxes). Mỗi hộp gồm các thông số: tâm x, y; chiều rộng w, chiều cao h (thường được chuẩn hóa), + một "điểm tin cậy (confidence score)" biểu thị: khả năng có đối tượng + độ tin cậy vị trí. Ngoài ra, mỗi ô cũng dự đoán xác suất (conditional class probabilities) cho mỗi lớp đối tượng mà ảnh đó có thể chứa.

- **Kết hợp kết quả và loại bỏ dư thừa (Non-Maximum Suppression – NMS):** Vì nhiều hộp có thể "đụng chồng" nhau (ví dụ cùng dự đoán một đối tượng), sau khi mạng đưa ra tất cả dự đoán, YOLO sử dụng kỹ thuật NMS để giữ lại hộp có "điểm tin cậy tốt nhất" và loại bỏ các hộp dư chồng lặp. Kết quả cuối: một danh sách các hộp (bounding boxes), mỗi hộp có nhãn lớp đối tượng và độ tin cậy — tương ứng với những đối tượng được phát hiện trong ảnh.



Hình 2.7.14: Cơ chế hoạt động của YOLO

2.7.2 Hàm mất mát (Loss Function) trong YOLO

Trong YOLO, hàm mất mát (loss function) được xây dựng từ sự khác biệt giữa dự đoán của mô hình và nhãn thực tế. Tổng độ lỗi là sự kết hợp của ba thành phần chính, mỗi thành phần phản ánh một khía cạnh quan trọng trong quá trình phát hiện đối tượng:

1. **Classification Loss - Sai số phân loại:** Đo mức độ chính xác khi mô hình dự đoán lớp của đối tượng trong mỗi ô lưới. Mục tiêu là đảm bảo mô hình không chỉ phát hiện được vật thể, mà còn nhận diện đúng loại của nó.

2. **Localization Loss - Sai số định vị bounding box:** Đánh giá độ lệch giữa bounding box dự đoán và bounding box thật dựa trên bốn tham số: vị trí tâm (x, y) và kích thước (w, h). Thành phần này giúp mô hình học cách khoanh vùng đối tượng chính xác hơn.

3. **Confidence Loss - Sai số về mức độ tin cậy:** Phản ánh sự khác biệt giữa "mức độ chắc chắn" mà mô hình cho rằng ô lưới chứa một vật thể và giá trị nhãn thực tế. Đây là yếu tố quan trọng giúp YOLO biết được ô nào nên dự đoán và ô

nào nên bỏ qua.

Classification Loss:

Classification loss là sai số phản ánh mức độ chính xác khi mô hình dự đoán lớp của đối tượng. Thành phần này chỉ được tính cho những ô lưới thật sự chứa object, còn các ô không có vật thể sẽ được bỏ qua để tránh làm nhiễu quá trình học. Classification loss được xác định theo công thức sau:

$$L_{classification} = \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \tag{2.7.9}$$

Trong đó:

- I_i^{obj}: Biến chỉ báo, bằng 1 nếu ô lưới i chứa object, ngược lại bằng 0.
- $\hat{p}_i(c)$: Xác xuất có điều kiện của lớp c tại ô vuông tương ứng mà mô hình dự đoán.

Localization Loss:

Localization loss là thành phần đo sai số vị trí và kích thước của bounding box mà mô hình dự đoán. Nó so sánh tọa độ tâm cùng chiều rộng và chiều cao của bounding box dự đoán với giá trị thực tế (ground truth) trong dữ liệu huấn luyện. Một điểm quan trọng là các giá trị này không được tính trực tiếp theo kích thước ảnh gốc, mà phải được chuẩn hoá về khoảng [0, 1] dựa trên kích thước của ô lưới và vị trí tương đối của bounding box. Việc chuẩn hóa giúp mô hình học ổn định hơn, hội tụ nhanh hơn và đạt độ chính xác cao hơn. Localization loss được tính bằng tổng sai số giữa độ tâm (x,y) và kích thước (w,h) của bounding box dự đoán so với ground truth. Ở mỗi ô lưới chứa đối tượng, YOLO chọn một bounding box có IOU cao nhất với ground truth để chịu trách nhiệm dự đoán. Sai số được tính dựa trên bounding box được chọn này. Công thức tính Localization loss như sau:

$$L_{localization} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \tag{2.7.10}$$

Trong đó:

- $_{coord}$: Hệ số điều chỉnh, thường được đặt giá trị cao hơn để nhấn mạnh tầm quan trọng của localization loss.
- 1_{ij}^{obj}: Biến chỉ báo, bằng 1 nếu ô lưới i chứa object và bounding box j chịu trách nhiệm dự đoán, ngược lại bằng 0.
- x_i, y_i, w_i, h_i: Tọa độ tâm và kích thước thực tế của bounding box.
- $\hat{x_i}, \hat{y_i}, \hat{w_i}, \hat{h_i}$: Tọa độ tâm và kích thước dự đoán của bounding box.

Confidence Loss:

Confidence loss đo mức độ chênh lệch giữa "độ tin cậy" mà mô hình gán cho một bounding box (khả năng ô lưới đó chứa object) và giá trị nhãn thực tế. Khác với hai thành phần loss còn lại, confidence loss được tính cho tất cả các ô lưới, bao gồm cả ô có đối tượng và ô không có đối tượng. Nhờ đó, mô hình học được cách phân biệt đâu là vùng chứa object thật và đâu là vùng nền (background), giúp giảm dự đoán sai và tăng độ chính xác tổng thể.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \right] \tag{2.7.11}$$

Trong đó:

- λ_{noobj}: Hệ số điều chỉnh quan trọng được sử dụng để kiểm soát trọng số của phần lỗi liên quan đến các bounding box không chứa đối tượ (no-object).
- 1_{ij}^{noobj}: Cho biết bounding box thứ của cell i không chứa đối tượng.
- C_i: Độ tin cậy thực tế (1 nếu có đối tượng, 0 nếu không có).
- $\hat{C}_i$: Độ tin cậy dự đoán của bounding box.

Tổng hợp hàm mất mát trong YOLO:

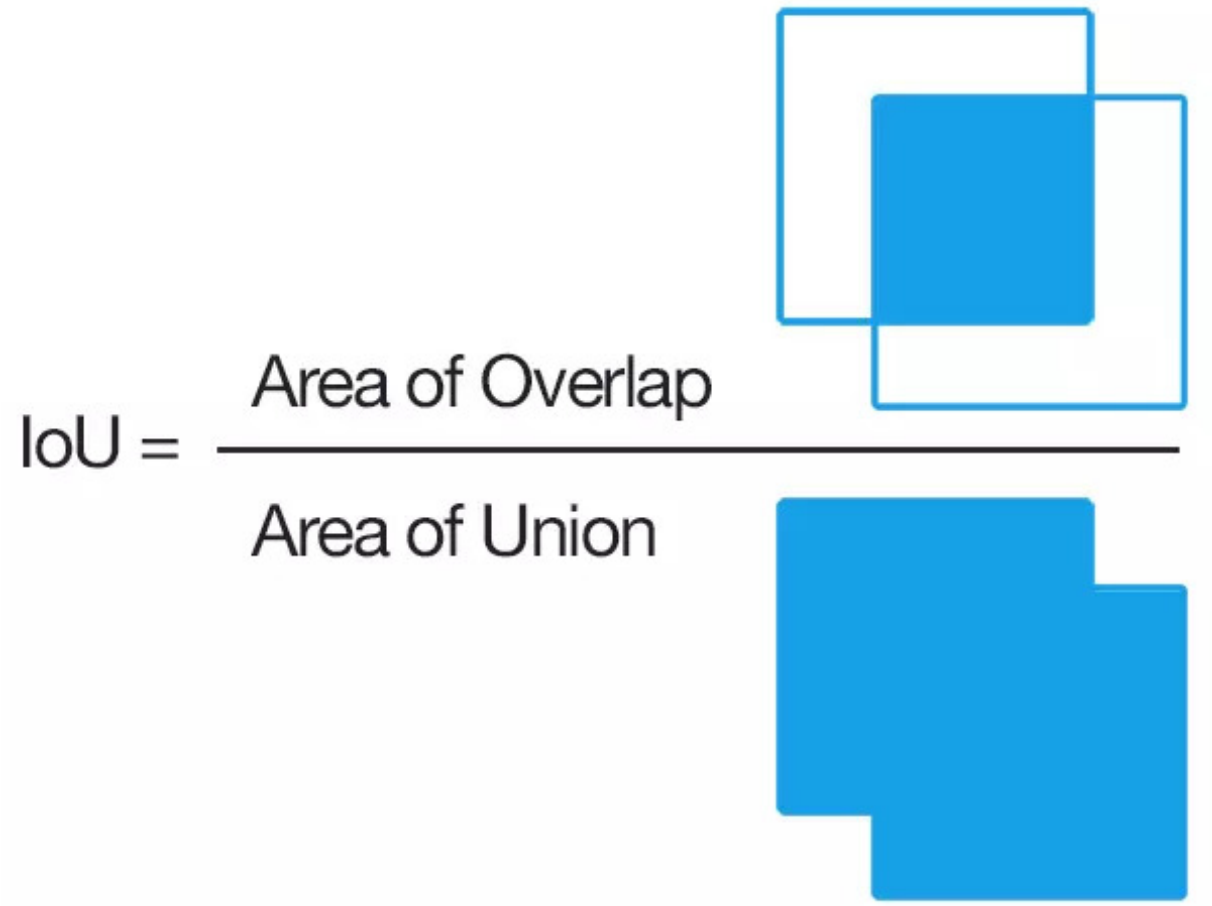
$$L_{total} = L_{classification} + L_{localization} + L_{confidence} \tag{2.7.12}$$

2.7.3 Các chỉ số đánh giá hiệu suất của YOLO

Trong bài toán phát hiện đối tượng, ba chỉ số quan trọng thường được sử dụng để đánh giá hiệu suất mô hình gồm Mean Average Precision (mAP), Average Precision (AP) và Intersection over Union (IoU). **Mean Average Precision (mAP):** mAP đánh giá hiệu suất tổng thể của mô hình bằng cách lấy trung bình giá trị AP trên toàn bộ các lớp đối tượng. Để tính mAP, trước hết ta xác định AP của từng lớp dựa trên tích phân của đường cong Precision–Recall. Sau đó, mAP được tính bằng trung bình cộng của các AP này. Giá trị mAP càng cao chứng tỏ mô hình đạt được độ chính xác (precision) và độ thu hồi (recall) tốt trên toàn bộ tập đối tượng. **Average Precision (AP):** AP phản ánh hiệu suất của mô hình tại các mức độ thu hồi khác nhau thông qua quan hệ giữa hai đại lượng:

- Precision (Độ chính xác): TP/(TP + FP)
- Recall (Độ thu hồi): TP/(TP + FN)

AP được tính bằng diện tích dưới đường cong Precision-Recall (PR curve), thể hiện sự đánh đổi giữa khả năng phát hiện đúng (precision) và khả năng tìm được nhiều đối tượng nhất (recall). **Intersection over Union (IoU):** IoU đo mức độ trùng khớp giữa bounding box dự đoán và bounding box ground truth. Chỉ số này được tính bằng tỉ lệ giữa diện tích vùng giao nhau và diện tích vùng hợp nhất của hai bounding box. Giá trị IoU càng cao cho thấy mô hình định vị đối tượng càng chính xác. Công thức tính như hình 2.5.8 dưới đây:



Hình 2.7.15: Cách tính chỉ số IOU

2.7.4 NON-MAXIMUM SUPPRESSION

Trong giai đoạn suy luận, YOLO thường tạo ra nhiều bounding box trùng lặp, đặc biệt tại những khu vực có mật độ đối tượng cao hoặc khi các ô lưới lân cận cùng dự đoán một vật thể. Việc xuất hiện quá nhiều bounding box chồng chéo không chỉ gây dư thừa mà còn làm giảm chất lượng dự đoán. Để khắc phục vấn đề này, YOLO sử dụng kỹ thuật Non-Maximum Suppression (NMS). Phương pháp này sàng lọc và loại bỏ các bounding box kém quan trọng, chỉ giữ lại hộp có độ tin cậy cao nhất trong số các hộp chồng lặp lên nhau. Nhờ đó, mô hình tránh được việc "đếm" một đối tượng nhiều lần và tập trung vào các dự đoán chính xác nhất, giúp kết quả nhận dạng trở nên rõ ràng và đáng tin cậy hơn.

Hình 2.7.16: Kết quả sau khi áp dụng Non-Maximum Suppression

Các bước của Non-Maximum Suppression bao gồm:

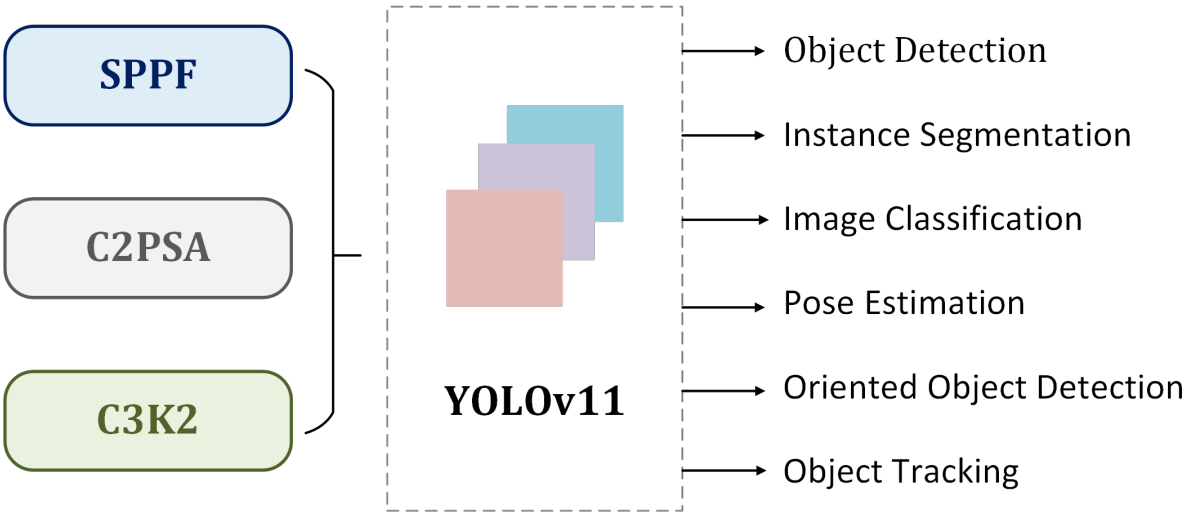
- Bước 1: Loại bỏ các bounding box có xác suất chứa vật thể nhỏ hơn một ngưỡng đã cho, thường là 0.5.
- Bước 2: Chọn bounding box có độ tin cậy cao nhất.
- Bước 3: Tính toán IoU giữa bounding box đã chọn và các bounding box còn lại. Loại bỏ những hộp có IoU lớn hơn một ngưỡng nhất định (ví dụ 0.4) để tránh chồng lấp.

2.8 TỔNG QUAN VỀ MÔ HÌNH YOLOV11

YOLO11 là thế hệ mới nhất trong chuỗi mô hình YOLO do Ultralytics phát triển, hướng tới bài toán phát hiện đối tượng theo thời gian thực với hiệu năng vượt trội. Kế thừa và mở rộng những thành tựu của các phiên bản tiền nhiệm, YOLO11 được trang bị nhiều cải tiến quan trọng về kiến trúc mạng cũng như chiến lược huấn luyện, qua đó nâng cao đồng thời độ chính xác, tốc độ suy luận và hiệu quả tính toán. Nhờ tính linh hoạt và khả năng thích ứng cao, YOLO11 phù hợp với nhiều bài toán thị giác máy tính trong các kịch bản ứng dụng thực tế.

2.8.1 Kiến trúc mô hình YOLOv11

Kế thừa nền tảng kiến trúc đã được khẳng định, YOLOv11 tiếp tục phát triển và hoàn thiện những thành quả của YOLOv8 thông qua các cải tiến về cấu trúc mạng và chiến lược tối ưu tham số, từ đó nâng cao đáng kể hiệu quả và độ chính xác trong bài toán phát hiện đối tượng.



Hình 2.8.17: Sơ đồ kiến trúc mạng YOLOv11

Các khối trong kiến trúc YOLOv11 bao gồm:

- **SPFF (Spatial Pyramid Pooling - Fast):** Khối SPPF được sử dụng để mở rộng vùng cảm thụ (receptive field) của mạng bằng cách áp dụng các phép pooling với kích thước khác nhau trên cùng một đặc trưng đầu vào. Nhờ đó, mô hình có thể khai thác thông tin ngữ cảnh đa tỷ lệ mà không làm tăng đáng kể chi phí tính toán, giúp cải thiện khả năng phát hiện các đối tượng có kích thước khác nhau.
- **C2PSA (Cross Stage Partial with Self-Attention):** C2PSA là khối đặc trưng kết hợp giữa cơ chế Cross Stage Partial (CSP) và Self-Attention. Khối này giúp tăng cường khả năng học mối quan hệ không gian – ngữ nghĩa giữa các vùng trong ảnh, đồng thời giảm số lượng tham số và chi phí tính toán. Nhờ đó, mô hình có thể tập trung tốt hơn vào các vùng quan trọng của đối tượng.
- **C3K2:** C3K2 là một biến thể của khối C3, sử dụng các lớp tích chập với kernel kích thước nhỏ (ví dụ 3x3) và cấu trúc residual để trích xuất đặc trưng hiệu quả.

Khối này giúp cân bằng giữa độ sâu mạng, khả năng biểu diễn đặc trưng và tốc độ suy luận, đặc biệt phù hợp cho các tác vụ phát hiện đối tượng thời gian thực.

- **Các khối đặc trưng nhiều màu ở trung tâm hình:** Các khối này biểu diễn các feature maps ở nhiều mức không gian khác nhau, tương ứng với các tầng đặc trưng đa tỷ lệ được trích xuất từ backbone/neck. Việc kết hợp các feature maps này cho phép mô hình phát hiện hiệu quả cả đối tượng nhỏ, trung bình và lớn.

2.8.2 Tính năng nổi bật của mô hình YOLOv11

YOLOv11 được phát triển như một bước tiến quan trọng trong dòng mô hình YOLO, tập trung đồng thời vào độ chính xác, tốc độ xử lý và hiệu quả tính toán. Những đặc điểm nổi bật của mô hình có thể được tóm lược như sau:

- **Trích xuất đặt trưng nâng cao:** YOLOv11 áp dụng kiến trúc backbone và neck được cải tiến, cho phép khai thác đặc trưng hình ảnh ở nhiều mức không gian và ngữ nghĩa khác nhau. Nhờ đó, mô hình nâng cao khả năng biểu diễn đặc trưng, đặc biệt hiệu quả trong việc phát hiện các đối tượng nhỏ, chồng lấp hoặc xuất hiện trong bối cảnh phức tạp.

- **Tối ưu hóa hiệu suất và tốc độ suy luận:** Với thiết kế kiến trúc tinh gọn cùng quy trình huấn luyện được tối ưu hóa, YOLOv11 đạt tốc độ xử lý vượt trội trong khi vẫn duy trì sự cân bằng hợp lý giữa độ chính xác và hiệu năng. Điều này giúp mô hình đáp ứng tốt các yêu cầu của các ứng dụng thời gian thực.

- **Độ chính xác cao với số lượng tham số giảm:** Nhờ các cải tiến trong thiết kế mô hình, YOLOv11m đạt giá trị mAP cao hơn trên bộ dữ liệu COCO, đồng thời giảm khoảng 22% số lượng tham số so với YOLOv8m. Sự tối ưu này giúp nâng cao hiệu quả tính toán, giảm yêu cầu tài nguyên mà không làm suy giảm chất lượng dự đoán.

- **Khả năng thích ứng linh hoạt trong nhiều môi trường triển khai:** YOLOv11 có thể được triển khai hiệu quả trên nhiều nền tảng khác nhau, từ các thiết bị biên (edge devices), hệ thống nhúng, đến các môi trường đám mây và hạ tầng tăng tốc

GPU của NVIDIA. Tính linh hoạt này giúp mô hình dễ dàng tích hợp vào các hệ thống ứng dụng thực tế.

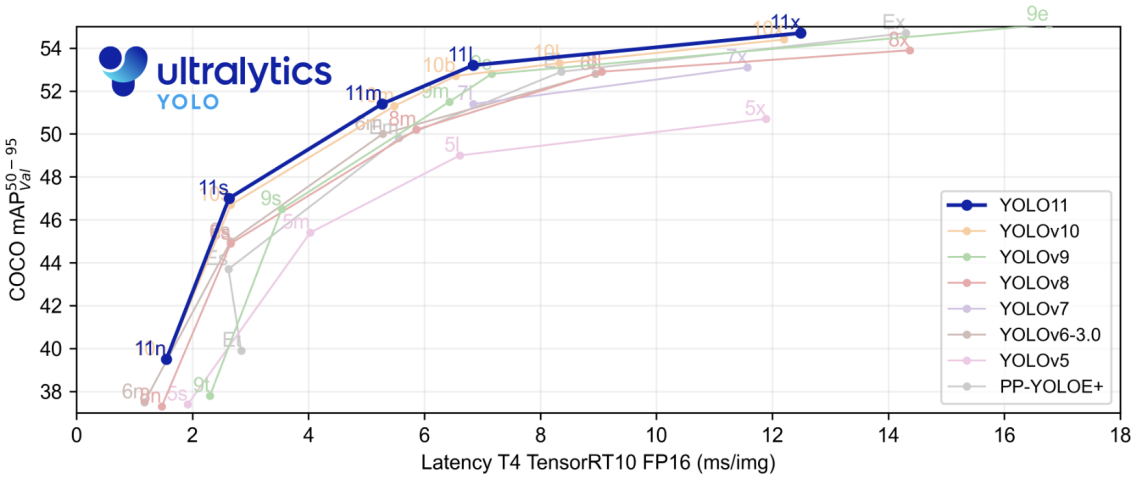
- **Hỗ trợ đa dạng các tác vụ thị giác máy tính:** Không chỉ giới hạn ở bài toán phát hiện đối tượng, YOLOv11 còn được thiết kế để hỗ trợ nhiều nhiệm vụ thị giác máy tính khác nhau như phân đoạn ảnh, phân loại hình ảnh, ước tính tư thế, phát hiện đối tượng theo hướng (oriented object detection). Nhờ đó, phạm vi ứng dụng của YOLOv11 được mở rộng đáng kể trong các bài toán thực tiễn.

Hiệu suất của mô hình YOLOv11 YOLOv11 - phiên bản mới nhất trong họ mô hình YOLO - đã được Ultralytics tiến hành đánh giá và so sánh hiệu suất với các phiên bản tiền nhiệm từ YOLOv5 đến YOLOv10. Kết quả thực nghiệm cho thấy YOLOv11 đạt được những cải tiến rõ rệt cả về độ chính xác lẫn tốc độ suy luận, qua đó khẳng định ưu thế vượt trội của kiến trúc mới.

Cụ thể, YOLOv11x đạt giá trị khoảng 54,5% $mAP50_{50-95}$ với độ trễ suy luận chỉ 13 ms, vượt qua toàn bộ các phiên bản YOLO trước đó trong cả hai tiêu chí độ chính xác và hiệu năng. Điều này cho thấy mô hình có khả năng xử lý các bài toán phát hiện đối tượng phức tạp trong thời gian thực với độ tin cậy cao.

Đối với YOLOv11m, mô hình mang lại mức độ chính xác tương đương hoặc tiệm cận với các biến thể kích thước lớn của các thế hệ YOLO trước, nhưng yêu cầu tài nguyên tính toán thấp hơn đáng kể. Đặc điểm này giúp YOLOv11m trở thành lựa chọn cân bằng giữa hiệu năng và chi phí triển khai.

Trong khi đó, YOLOv11s hướng đến các hệ thống yêu cầu độ trễ cực thấp. Mô hình đạt khoảng 47% mAP_{50-95} trong khoảng độ trễ từ 2-6 ms, cho phép triển khai hiệu quả trong các ứng dụng thời gian thực trên thiết bị biên mà vẫn duy trì mức độ chính xác chấp nhận được. Nhìn chung, các kết quả đánh giá cho thấy YOLOv11 không chỉ cải thiện hiệu suất so với các phiên bản trước, mà còn mở rộng khả năng ứng dụng trong nhiều kịch bản khác nhau, từ hệ thống nhúng hạn chế tài nguyên đến các nền tảng tính toán hiệu năng cao.



Hình 2.8.18: Hiệu suất của mô hình yolov11 so với các phiên bản trước

Ngoài ra, đường cong cải tiến của YOLOv11 cho thấy khả năng mở rộng (scaling) vượt trội giữa các biến thể của mô hình, cho phép khai thác tài nguyên tính toán một cách hiệu quả và linh hoạt hơn so với các thế hệ trước. Đặc tính này giúp các phiên bản YOLOv11 duy trì sự cân bằng tối ưu giữa độ chính xác và hiệu năng khi thay đổi quy mô mô hình. Những kết quả đạt được đã khẳng định YOLOv11 là một bước tiến đáng kể trong lĩnh vực phát hiện đối tượng theo thời gian thực, đáp ứng hiệu quả các yêu cầu ngày càng khắt khe của các ứng dụng thị giác máy tính hiện đại.

2.9 TỔNG QUAN VỀ MÔ PHỎNG GIAO THÔNG ĐÔ THỊ

2.9.1 Vai trò của mô phỏng trong quản lý giao thông đô thị

Sự gia tăng nhanh chóng của phương tiện cá nhân, quá trình đô thị hóa mạnh mẽ cùng với hạ tầng giao thông hạn chế đã khiến bài toán quản lý và điều hành giao thông đô thị trở nên ngày càng phức tạp. Các quyết định điều chỉnh tín hiệu đèn, tổ chức làn đường hay phân luồng giao thông nếu được triển khai trực tiếp ngoài thực tế thường tiềm ẩn rủi ro cao, chi phí lớn và khó đánh giá trước hiệu quả. Trong bối cảnh đó, mô phỏng giao thông đóng vai trò như một công cụ hỗ trợ quan trọng, cho phép tái hiện và phân

tích hành vi giao thông trong môi trường ảo trước khi áp dụng vào thực tiễn.

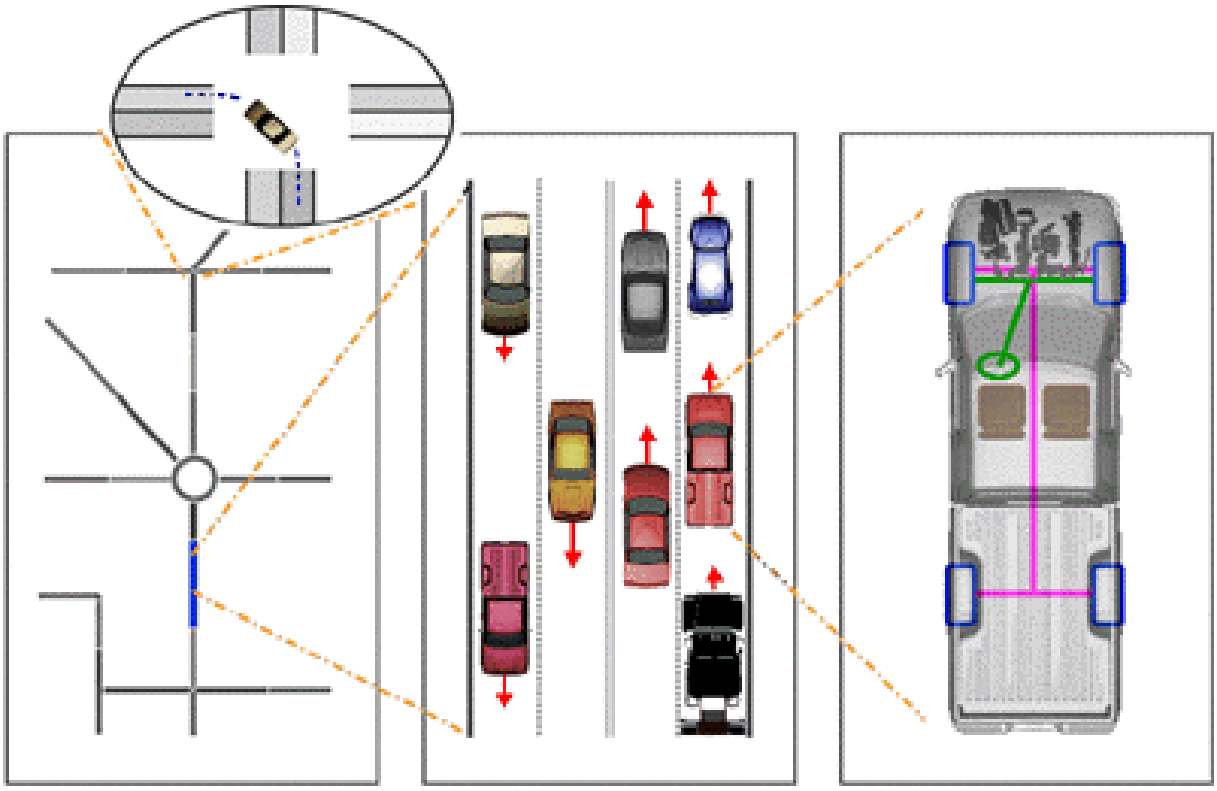
Mô phỏng giao thông đô thị cho phép các nhà nghiên cứu và cơ quan quản lý đánh giá tác động của nhiều kịch bản khác nhau, chẳng hạn như thay đổi lưu lượng xe, điều chỉnh chu kỳ tín hiệu đèn hoặc áp dụng các chiến lược điều tiết mới. Thông qua mô phỏng, các chỉ số quan trọng như thời gian lưu thông trung bình, mật độ xe, mức độ ùn tắc hay số lần dừng chờ có thể được định lượng một cách khách quan. Nhờ đó, mô phỏng không chỉ giúp giảm thiểu rủi ro khi triển khai ngoài thực tế mà còn góp phần nâng cao hiệu quả ra quyết định trong quản lý giao thông đô thị.

2.9.2 Khái niệm và phân loại mô phỏng giao thông

Mô phỏng giao thông là quá trình xây dựng một mô hình toán học hoặc mô hình tính toán nhằm mô tả và tái hiện sự chuyển động của các phương tiện, người tham gia giao thông và tương tác của chúng trên mạng lưới đường bộ theo thời gian. Tùy theo mức độ chi tiết và phạm vi mô phỏng, các mô hình giao thông thường được phân loại thành ba nhóm chính: mô phỏng vĩ mô, mô phỏng trung mô và mô phỏng vi mô.

Mô phỏng vĩ mô xem giao thông như một dòng chảy liên tục, tương tự chất lưu, trong đó các đặc trưng như lưu lượng, mật độ và tốc độ trung bình được mô hình hóa thông qua các phương trình toán học. Cách tiếp cận này phù hợp cho phân tích ở quy mô lớn nhưng hạn chế trong việc mô tả hành vi chi tiết của từng phương tiện.

Mô phỏng trung mô nằm giữa hai cấp độ vĩ mô và vi mô, trong đó phương tiện được nhóm thành các cụm hoặc dòng xe, cho phép cân bằng giữa độ chính xác và chi phí tính toán. Trong khi đó, mô phỏng vi mô mô tả chi tiết hành vi của từng phương tiện riêng lẻ, bao gồm gia tốc, giảm tốc, chuyển làn và tương tác với tín hiệu giao thông. Ngoài ra để chi tiết hơn, để mô tả hành vi bên trong của phương tiện và người lái ở mức liên tục theo thời gian, mô phỏng dưới vi mô được sử dụng. Mô phỏng dưới vi mô quan tâm đến các yếu tố như động cơ, hộp số, Mô-men xoắn, lực kéo, mức tiêu thụ nhiên liệu, phát thải khí CO2. [**?**]



Hình 2.9.19: Các mức độ chi tiết trong trong các mô hình mô phỏng giao thông khác nhau. Từ trái sang: vĩ mô, vi mô, dưới vi mô (trong vòng tròn là trung mô)

2.9.3 Mô phỏng giao thông theo thời gian và dựa trên dữ liệu

Một đặc điểm quan trọng của giao thông đô thị là tính động và biến thiên theo thời gian. Lưu lượng xe, tốc độ và mức độ ùn tắc có thể thay đổi đáng kể theo từng khung giờ trong ngày, theo điều kiện thời tiết hoặc theo các sự kiện bất thường. Do đó, các mô hình mô phỏng giao thông hiện đại thường được xây dựng dưới dạng mô phỏng theo thời gian rời rạc hoặc liên tục, cho phép cập nhật trạng thái của hệ thống tại mỗi bước thời gian.

Bên cạnh đó, xu hướng phát triển của các hệ thống giao thông thông minh đã thúc đẩy việc tích hợp dữ liệu thực vào mô phỏng. Các nguồn dữ liệu như camera giao thông, cảm biến hoặc dữ liệu lịch sử cho phép hiệu chỉnh mô hình mô phỏng sao cho phản ánh sát hơn điều kiện thực tế. Trong nhiều nghiên cứu, dữ liệu hình ảnh từ camera giao thông được xử lý để trích xuất các thông số như lưu lượng xe, mật độ hoặc tốc độ trung bình, sau đó được sử dụng làm đầu vào cho mô phỏng. Cách tiếp cận này giúp thu hẹp khoảng

cách giữa mô hình mô phỏng và tình trạng giao thông ngoài thực tế.

2.9.4 Ứng dụng của mô phỏng giao thông trong nghiên cứu và thực tiễn

Mô phỏng giao thông đô thị được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ quy hoạch hạ tầng, đánh giá tác động của các dự án giao thông đến hỗ trợ điều hành giao thông hàng ngày. Trong nghiên cứu học thuật, mô phỏng được sử dụng để kiểm chứng các giả thuyết, so sánh hiệu quả của các thuật toán điều khiển tín hiệu hoặc đánh giá các mô hình dự báo giao thông.

Trong thực tiễn, mô phỏng giao thông đóng vai trò như một "phòng thí nghiệm ảo", nơi các kịch bản điều chỉnh có thể được thử nghiệm trước khi triển khai. Đặc biệt, khi kết hợp với các mô hình học máy và học sâu, mô phỏng không chỉ dừng lại ở việc tái hiện hiện trạng mà còn có khả năng hỗ trợ dự báo ngắn hạn và đánh giá các chiến lược điều chỉnh tín hiệu đèn trong tương lai. Điều này mở ra tiềm năng lớn cho việc xây dựng các hệ thống hỗ trợ ra quyết định trong quản lý giao thông đô thị.

2.10 CÔNG CỤ MÔ PHỎNG GIAO THÔNG: SUMO

2.10.1 Giới thiệu phần mềm SUMO

SUMO (Simulation of Urban Mobility) là phần mềm mô phỏng giao thông đô thị miễn phí và mã nguồn mở, được phát triển bởi Viện Nghiên cứu Giao thông Vận tải Đức (German Aerospace Center - DLR). Phần mềm này cho phép xây dựng và mô phỏng các mạng lưới giao thông đô thị phức tạp với mức độ chi tiết cao, bao gồm hệ thống đường bộ, nút giao thông, làn đường, hệ thống tín hiệu đèn giao thông, nhiều loại phương tiện khác nhau (ô tô, xe máy, xe buýt, xe tải) cũng như các đối tượng tham gia giao thông khác như người đi bộ và xe đạp. SUMO hoạt động dựa trên mô hình mô phỏng vi mô, trong đó mỗi phương tiện được mô phỏng như một thực thể độc lập, cho phép theo dõi chính xác vị trí, vận tốc và hành vi di chuyển của từng đối tượng trong suốt quá trình mô phỏng.

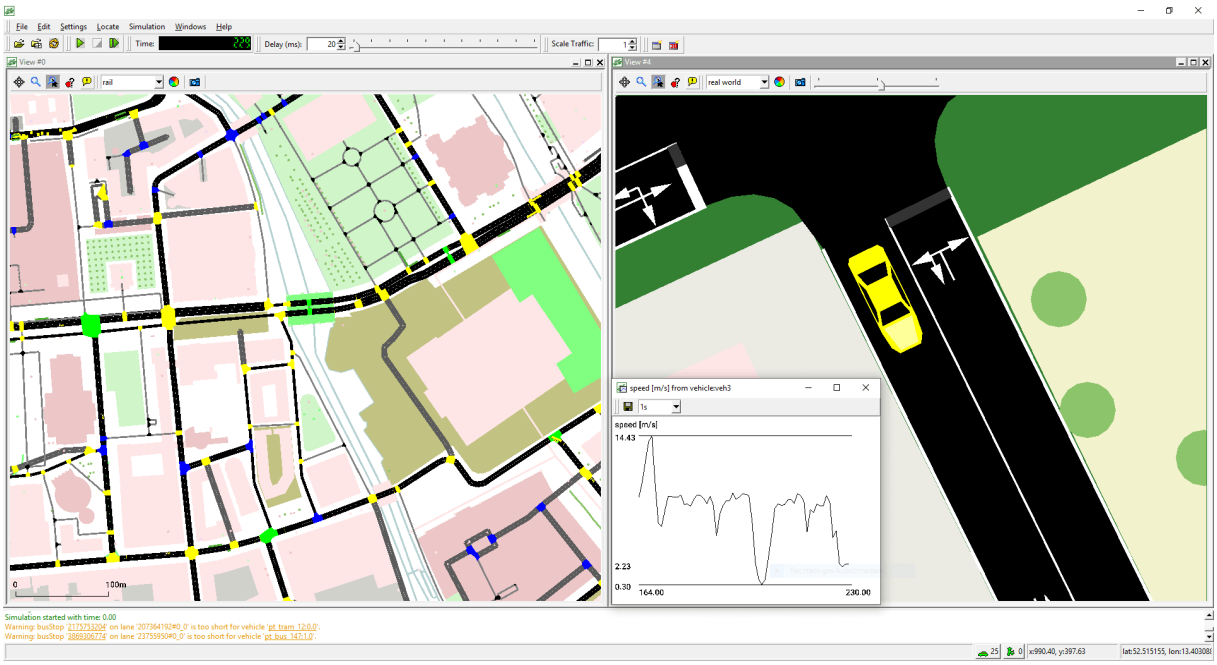
Hình 2.10.20: Phần mềm SUMO

Nhờ khả năng tùy biến cao thông qua các tham số mô phỏng, mô hình hành vi lái xe và dữ liệu đầu vào linh hoạt, SUMO được sử dụng rộng rãi trong nghiên cứu khoa học, quy hoạch giao thông và kỹ thuật giao thông. Một số ứng dụng tiêu biểu của SUMO bao gồm:

- **Phân tích và hiểu rõ hoạt động của mạng lưới giao thông:** SUMO cho phép mô phỏng chi tiết hành vi của các phương tiện và người tham gia giao thông trong mạng lưới đô thị, từ đó giúp người dùng hiểu rõ hơn cơ chế vận hành của hệ thống giao thông. Thông qua mô phỏng, các điểm nghẽn, khu vực dễ xảy ra ùn tắc và những vấn đề tiềm ẩn trong tổ chức giao thông có thể được xác định một cách trực quan và định lượng, làm cơ sở cho việc đề xuất các giải pháp cải thiện.
- **Hỗ trợ lập kế hoạch và thiết kế giao thông đô thị:** Phần mềm được sử dụng để xây dựng và đánh giá các kịch bản quy hoạch giao thông, bao gồm mở rộng hoặc điều chỉnh mạng lưới đường bộ, tối ưu hóa hệ thống đèn tín hiệu, tổ chức lại luồng giao thông, cũng như triển khai và đánh giá hiệu quả của các hệ thống giao thông công cộng.
- **Đánh giá tác động của các dự án và chính sách giao thông:** SUMO cho phép mô phỏng và so sánh các kịch bản trước và sau khi triển khai dự án giao thông, từ đó đánh giá tác động của các biện pháp can thiệp đối với lưu lượng, mật độ, thời gian di chuyển và mức độ ùn tắc. Điều này giúp các nhà quản lý và kỹ sư giao thông đưa ra quyết định dựa trên dữ liệu mô phỏng thay vì chỉ dựa vào kinh nghiệm.

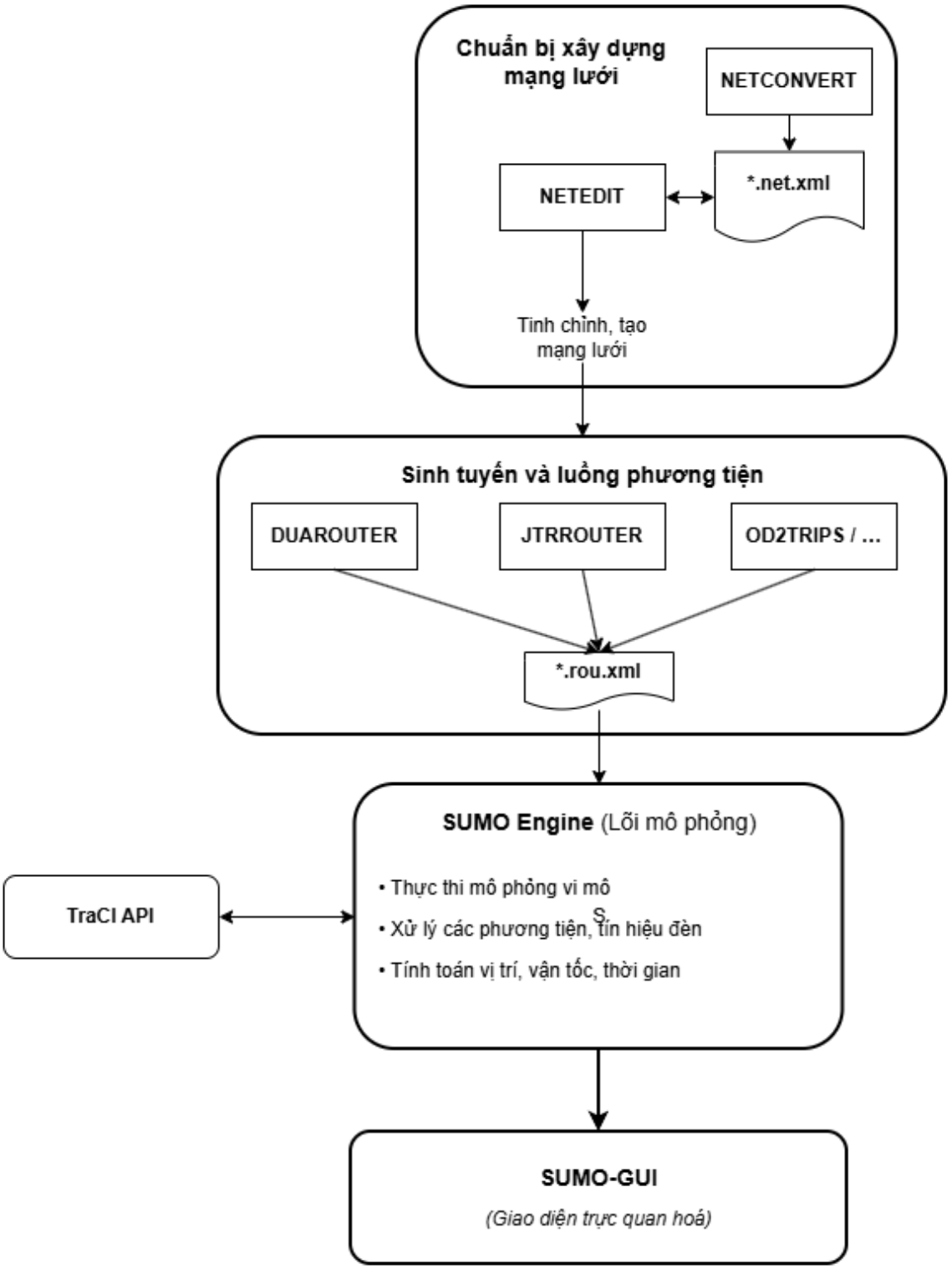
- **Hỗ trợ nghiên cứu và thử nghiệm các giải pháp giao thông thông minh:** Nhờ khả năng tích hợp với các thuật toán điều khiển và mô hình học máy, SUMO thường được sử dụng trong nghiên cứu các hệ thống giao thông thông minh (ITS), chẳng hạn như điều khiển đèn tín hiệu thích nghi, dự báo mật độ giao thông, hay đánh giá tác động của phương tiện tự hành trong môi trường giao thông đô thị.

- **Giảm chi phí và rủi ro khi triển khai thực tế:** Việc thử nghiệm các phương án tổ chức và quản lý giao thông trong môi trường mô phỏng giúp giảm thiểu chi phí, rủi ro và tác động tiêu cực có thể xảy ra khi áp dụng trực tiếp các giải pháp mới vào hệ thống giao thông thực tế.



Hình 2.10.21: Giao diện mô phỏng của SUMO

2.10.2 Kiến trúc tổng thể của SUMO



Hình 2.10.22: Kiến trúc tổng thể của SUMO

Trong đó:

- **Lớp chuẩn bị xây dựng mạng lưới:** NETEDIT được dùng để xây dựng, chỉnh sửa hạ tầng giao thông một cách trực quan; NETCONVERT nhập/chuẩn hoá mạng lưới từ các dữ liệu như OSM, VISUM sang định dạng .net.xml chuẩn của SUMO.
- **Lớp sinh tuyến và luồng phương tiện:** DUAROUTER và JTRROUTER xử lý tạo các lộ trình di chuyển cho phương tiện dựa trên mạng lưới và dữ liệu yêu cầu; OD2TRIPS hoặc các công cụ tương tự sinh ra các bài toán O-D thành các chuyến xe riêng lẻ. Các tệp này (như .rou.xml) sẽ được SUMO sử dụng khi mô phỏng.
- **Lớp SUMO Engine:** Đây là thành phần trung tâm thực thi mô phỏng vi mô, xử lý chuyển động từng phương tiện khoảng theo thời gian, tính toán tương tác giữa các đối tượng, tín hiệu giao thông và các luật lưu thông.
- **Lớp tương tác và trực quan hóa:** Cho phép quan sát mô phỏng, xem phương tiện, mạng lưới và tín hiệu giao thông theo thời gian thực; nhưng GUI không ảnh hưởng đến logic mô phỏng gốc (chỉ để hiển thị).
- **Lớp giao diện lập trình TraCI:** TraCI (Traffic Control Interface) là giao diện lập trình cho phép ứng dụng bên ngoài (Python, C++, RL/AI...) kết nối và điều khiển mô phỏng đang chạy, truy vấn trạng thái đối tượng, điều chỉnh tín hiệu, phương tiện, hoặc thu thập dữ liệu theo thời gian thực. TraCI thường được dùng khi tích hợp AI/ML với SUMO.
- **Lớp giao diện lập trình SUMO-GUI:** Cho phép quan sát mô phỏng, xem phương tiện, mạng lưới và tín hiệu giao thông theo thời gian thực; nhưng GUI không ảnh hưởng đến logic mô phỏng gốc (chỉ để hiển thị).

2.10.3 Nguyên lý mô phỏng trong SUMO

SUMO (Simulation of Urban Mobility) là một phần mềm mô phỏng giao thông vi mô (microscopic), trong đó mỗi phương tiện và hành vi của nó được mô phỏng một cách riêng lẻ trên một mạng lưới đường đã được định nghĩa. Mô phỏng vi mô cho phép mô hình hóa các yếu tố chi tiết như vận tốc, vị trí, khoảng cách giữa các xe và phản ứng

của từng phương tiện đối với môi trường xung quanh, bao gồm đèn tín hiệu, mật độ lưu thông và tương tác với các phương tiện khác. Đây là điểm khác biệt so với mô phỏng vĩ mô, nơi lưu lượng được xem như dòng tổng thể mà không xét chi tiết từng đơn vị xe riêng lẻ.

Một trong những nguyên lý then chốt trong SUMO là mô hình car-following — mô hình hành vi lái xe theo sau. Trong mô hình này, mỗi phương tiện (được gọi là ego) điều chỉnh vận tốc dựa trên khoảng cách và vận tốc của phương tiện phía trước (leader), với mục tiêu tránh va chạm và duy trì khoảng cách an toàn. Trong SUMO, mô hình car-following được phát triển dựa trên công trình của Stefan Krauß và các mở rộng sau này, cho phép mô phỏng hành vi longitudinal (dọc theo chiều di chuyển) của xe một cách chi tiết.

Ngoài hành vi theo sau, SUMO cũng mô phỏng hành vi thay đổi làn và các quyết định liên quan đến chuyển làn. Các thuật toán lane-changing trong SUMO đánh giá mức lợi ích của việc chuyển sang làn khác — ví dụ như khả năng đạt được vận tốc cao hơn — và điều kiện an toàn khi thực hiện thao tác này, từ đó giả lập hành vi lái xe lateral (theo chiều ngang).

Mô phỏng trong SUMO diễn ra ở mức độ rời rạc theo thời gian: tại mỗi bước thời gian (time step), vị trí và vận tốc của từng phương tiện được tính toán dựa trên mô hình hành vi, đồng thời các sự kiện như tín hiệu đèn, thay đổi tuyến đường hay phản ứng với phương tiện khác được cập nhật. SUMO sử dụng phương pháp mô phỏng space-continuous (không gian liên tục) và time-discrete (thời gian rời rạc), cho phép mô phỏng chính xác sự di chuyển của phương tiện theo từng bước thời gian được chọn.

2.10.4 Điều khiển tín hiệu giao thông trong SUMO

Trong mô phỏng giao thông bằng SUMO, tín hiệu đèn giao thông (traffic lights) được xem như một thành phần điều khiển quan trọng tại các nút giao nhằm điều phối lưu lượng phương tiện và giảm thiểu xung đột giữa các hướng di chuyển. SUMO cho phép mô hình hóa chi tiết các chương trình tín hiệu, bao gồm các pha (phases), độ dài chu kỳ, thứ tự chuyển trạng thái và các trạng thái cụ thể của từng tín hiệu (đỏ, vàng, xanh). Những chương trình này có thể được khai báo trực tiếp trong mạng mô phỏng

hoặc được cung cấp dưới dạng tệp bổ sung để tùy chỉnh. SUMO hỗ trợ nhiều chế độ điều khiển tín hiệu khác nhau. Ở mức cơ bản, các tín hiệu có thể hoạt động theo chu kỳ cố định (static control), trong đó mỗi pha được định nghĩa rõ ràng với thời lượng cụ thể và tín hiệu sẽ lặp lại tuần hoàn trong suốt quá trình mô phỏng. Đây là dạng điều khiển truyền thống thường dùng trong các bài toán tham chiếu hoặc so sánh đơn giản. Ngoài ra, với giao diện điều khiển thời gian chạy TraCI (Traffic Control Interface), SUMO cho phép các chương trình điều khiển bên ngoài (ví dụ script Python) kết nối với mô phỏng để điều chỉnh tín hiệu động theo trạng thái mạng. Qua TraCI, có thể truy vấn trạng thái của mô phỏng và gửi các lệnh như thay đổi pha hiện tại, điều chỉnh thời lượng pha hay chuyển sang chương trình tín hiệu khác. Điều này mở ra khả năng tích hợp với các mô hình dự báo, thuật toán điều khiển máy học hoặc điều khiển thích ứng phức tạp hơn, nơi quyết định điều khiển được đưa ra bởi một bộ điều khiển bên ngoài dựa trên dữ liệu đầu vào thời gian thực mô phỏng.

2.10.5 Mô phỏng lưu lượng và tốc độ phương tiện biến thiên theo thời gian

None

2.11 Tổng quan về OpenStreetMap

OpenStreetMap (OSM) là một dự án bản đồ mở, được xây dựng theo mô hình cộng đồng, cho phép người dùng trên toàn thế giới cùng thu thập, chỉnh sửa và chia sẻ dữ liệu bản đồ địa lý một cách tự do. Mục tiêu chính của OpenStreetMap là cung cấp một nguồn dữ liệu không gian mở, miễn phí, có thể sử dụng và tái phân phối cho nhiều mục đích khác nhau, từ nghiên cứu học thuật, phát triển ứng dụng đến các hệ thống thông tin địa lý (GIS) và mô phỏng giao thông.

Dữ liệu của OpenStreetMap được tổ chức dưới dạng các đối tượng không gian cơ bản, bao gồm *node* (điểm), *way* (đường hoặc đa giác) và *relation* (mối quan hệ giữa các đối tượng). Mỗi đối tượng được gán các thuộc tính (tag) để mô tả đặc trưng thực tế như loại đường, số làn xe, giới hạn tốc độ, hướng lưu thông hoặc chức năng của công trình.

Nhờ cấu trúc linh hoạt này, OSM có khả năng biểu diễn chi tiết mạng lưới giao thông đô thị và các yếu tố hạ tầng liên quan.

Trong lĩnh vực mô phỏng giao thông, OpenStreetMap đóng vai trò là nguồn dữ liệu đầu vào quan trọng để xây dựng và tái hiện mạng lưới đường bộ thực tế. Các phần mềm mô phỏng như SUMO có thể chuyển đổi trực tiếp dữ liệu OSM sang mô hình mạng giao thông, bao gồm các nút giao, đoạn đường, làn xe và tín hiệu đèn giao thông. Việc sử dụng OSM giúp giảm đáng kể công sức xây dựng mạng mô phỏng thủ công, đồng thời nâng cao tính thực tế và khả năng mở rộng của hệ thống mô phỏng.

Với tính mở, độ phủ rộng và khả năng cập nhật liên tục, OpenStreetMap ngày càng được sử dụng phổ biến trong các nghiên cứu về giao thông thông minh, quy hoạch đô thị và hệ thống hỗ trợ ra quyết định. Trong phạm vi đề tài này, OSM được sử dụng như nền tảng dữ liệu không gian để xây dựng mạng lưới giao thông phục vụ cho mô phỏng và phân tích tình trạng giao thông đô thị.

Chương 3

THIẾT KẾ HỆ THỐNG

3.1 YÊU CẦU CỦA HỆ THỐNG

Hệ thống được đề xuất hướng tới việc mô phỏng, dự báo và điều tiết giao thông dựa trên dữ liệu hình ảnh snapshot thu thập từ camera giao thông, kết hợp các kỹ thuật học sâu và mô phỏng giao thông vi mô. Để đáp ứng mục tiêu nghiên cứu và đảm bảo khả năng triển khai thực tế, hệ thống cần thỏa mãn các yêu cầu chức năng và phi chức năng sau.

Về giai đoạn thu thập dữ liệu, hệ thống cần đảm bảo các yêu cầu sau:

- Tiếp nhận được dữ liệu hình ảnh giao thông dạng snapshot từ các camera giao thông đặt tại các nút giao.
- Các ảnh này phải được quản lý, lưu trữ và gắn nhãn thời gian rõ ràng.

Về giai đoạn tiền xử lý dữ liệu, nhận diện và đếm phương tiện, hệ thống cần đáp ứng các yêu cầu sau:

- Các ảnh đầu vào cần được thực hiện các bước xử lý ảnh cơ bản như tăng độ phân giải, giảm nhiễu và chuẩn hóa kích thước.
- Hệ thống tích hợp mô hình YOLOv11 nhằm nhận diện các phương tiện giao thông

chính trên từng ảnh snapshot, trong đó các phương tiện được quy ước và phân loại thành hai nhóm: xe hai bánh (xe máy) và xe bốn bánh (xe hơi).

- Kết quả phát hiện bao gồm số lượng phương tiện theo từng loại tại mỗi thời điểm và tại từng địa điểm giám sát cụ thể, trong đó "địa điểm" được hiểu là vị trí camera đại diện cho một khu vực giao thông trên bản đồ.
- Lưu trữ kết quả đếm phương tiện với timestamp và vị trí tương ứng để phục vụ cho các bước xử lý tiếp theo.
- Đảm bảo độ chính xác cao trong việc nhận diện và đếm phương tiện, với sai số không vượt quá 5% so với thực tế.

Tiếp đến là yêu cầu về mô phỏng giao thông:

- None

Đối với giai đoạn xây dựng chuỗi thời gian và dự báo lưu lượng giao thông, hệ thống cần thỏa mãn các yêu cầu sau:

- None

Đối với giai đoạn tích hợp và điều khiển luồng giao thông, hệ thống cần đáp ứng các yêu cầu sau:

- Hệ thống tích hợp các giá trị dự báo từ LSTM vào môi trường mô phỏng giao thông SUMO, trong đó lưu lượng phương tiện, tốc độ dòng xe hoặc phân bố phương tiện tại các nút giao được điều chỉnh tương ứng với trạng thái giao thông dự kiến.
- None

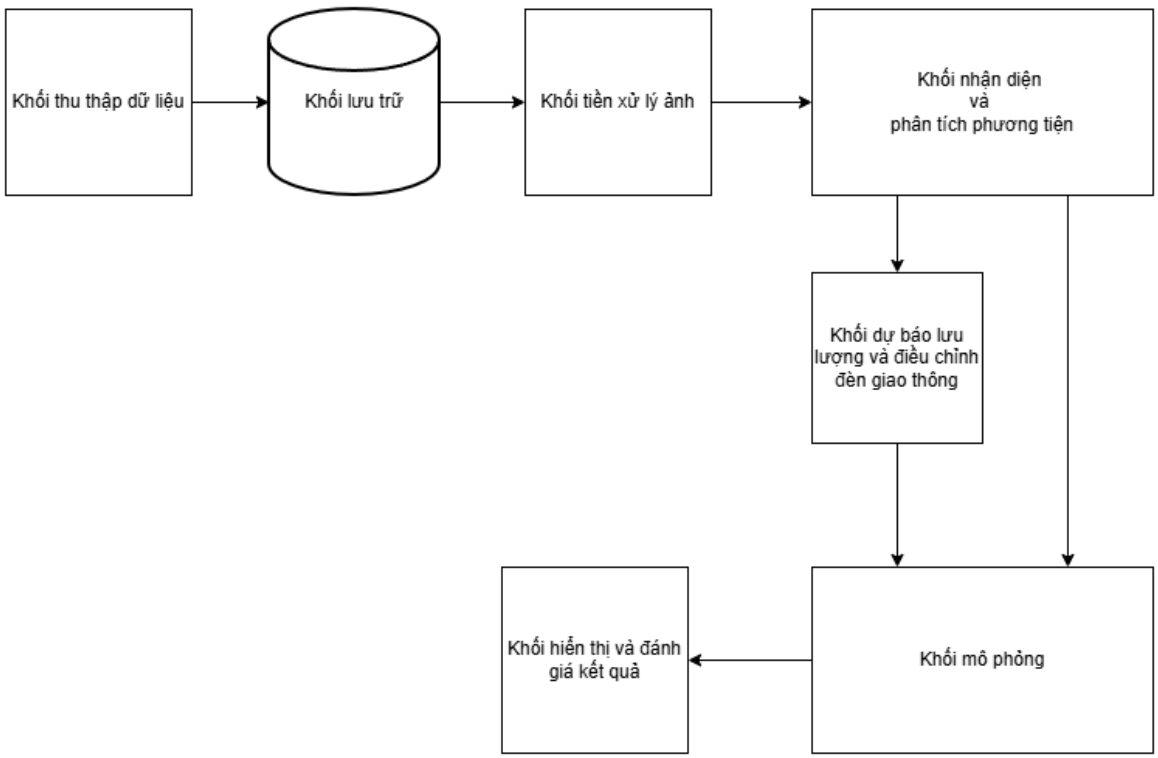
Cuối cùng, về giai đoạn trực quan hóa và phân tích, hệ thống cần:

- Cung cấp giao diện trực quan để hiển thị kết quả nhận diện, đếm phương tiện, dự báo lưu lượng và mô phỏng giao thông.

- Hỗ trợ các biểu đồ, bản đồ nhiệt và các công cụ phân tích để người dùng có thể dễ dàng hiểu và đánh giá tình hình giao thông.
- So sánh hiệu quả giữa việc xử dụng chiến lược điều tiết và không sử dụng để đánh giá hiệu quả của các biện pháp điều tiết dựa trên các chỉ số: thời gian di chuyển trung bình, thời gian đợi, chiều dài hàng đợi.

3.2 KIẾN TRÚC HỆ THỐNG

3.2.1 Sơ đồ khối hệ thống



Hình 3.2.1: Sơ đồ khối kiến trúc hệ thống

Chức năng từng khối:

- **Khối thu thập dữ liệu:** Khối thu thập dữ liệu chịu trách nhiệm tiếp nhận các hình ảnh snapshot từ nguồn mở Open Street Map (OSM). Trong đề tài này, dữ liệu không được thu thập trực tiếp từ hệ thống camera vật lý mà được lấy từ nguồn dữ

liệu mở, nơi các camera giao thông đã được thiết lập và công bố sẵn cho mục đích quan sát và tham khảo. Quá trình thu thập dữ liệu được thực hiện thông qua truy suất tự động (web scraping) để tải về các hình ảnh snapshot tại các thời điểm xác định (khoảng 12 giây một ảnh).

- **Khối lưu trữ:** Khối lưu trữ đóng vai trò quản lý và tổ chức toàn bộ dữ liệu hình ảnh thu thập được cũng như các dữ liệu trung gian phát sinh trong quá trình xử lý. Các ảnh snapshot thu thập từ camera phía OSM được lưu trữ dưới dạng cấu trúc thư mục phân cấp trên hệ thống lưu trữ tệp tin đám mây Google Drive. Mỗi ảnh được gắn nhãn thời gian rõ ràng và sắp xếp theo từng thư mục tương ứng với địa điểm giám sát, nhằm đảm bảo tính nhất quán và thuận tiện cho việc truy xuất, xử lý và phân tích về sau. Bên cạnh đó, các ảnh kết quả sau khi nhận diện và đếm phương tiện, dữ liệu thống kê mật độ giao thông theo thời gian, cũng như dữ liệu đầu vào và đầu ra của mô hình LSTM đều được lưu trữ tập trung tại khối này.

- **Khối tiền xử lý ảnh:** Khối tiền xử lý ảnh đảm nhiệm việc thực hiện các phép biến đổi cần thiết nhằm cải thiện chất lượng dữ liệu hình ảnh trước khi đưa vào mô hình nhận diện. Nguyên nhân là do các ảnh snapshot thu thập từ camera giao thông công khai trên mạng thường có chất lượng không đồng đều, chịu ảnh hưởng bởi nhiều yếu tố như độ phân giải thấp, nhiễu ảnh, điều kiện ánh sáng không ổn định và góc chụp chưa tối ưu. Do đó, khối này áp dụng một số kỹ thuật xử lý ảnh cơ bản, bao gồm tăng cường độ phân giải (super-resolution), lọc nhiễu (denoising), chuẩn hóa kích thước ảnh (resizing) và chia nhỏ ảnh nhằm đảm bảo dữ liệu đầu vào cho mô hình nhận diện đạt chất lượng tốt hơn và có tính nhất quán, dễ dàng nhận diện.

- **Khối nhận diện và phân tích phương tiện:** Khối nhận diện và phân tích phương tiện sử dụng mô hình học sâu YOLOv11 để thực hiện phát hiện và phân loại các phương tiện giao thông xuất hiện trong từng ảnh snapshot. Mô hình được khởi tạo từ trọng số huấn luyện sẵn (pre-trained) trên các tập dữ liệu lớn và đa dạng, nhờ đó có khả năng nhận diện hiệu quả các loại phương tiện phổ biến. Tuy nhiên, nhằm đơn giản hóa bài toán và phù hợp với mục tiêu nghiên cứu, các phương tiện được quy ước và gom nhóm thành hai lớp chính, bao gồm xe hai bánh (đại diện cho xe máy) và xe bốn bánh (đại diện cho xe hơi). Bên cạnh đó, do chất lượng

hình ảnh thu thập từ camera giao thông còn hạn chế và các bước tiền xử lý không thể khắc phục hoàn toàn các yếu tố bất lợi như góc quay cao, góc quay xiên hoặc mật độ giao thông lớn, một số phương tiện đặc biệt là xe máy có thể bị che khuất hoặc chồng chéo, dẫn đến độ chính xác nhận diện suy giảm. Để khắc phục vấn đề này, khối xử lý còn tích hợp các thuật toán xử lý ảnh bổ sung như Polygon Fill, Rectangle Fill, Pixel-wise Logic và Non-zero Pixel Count nhằm tính toán tỷ lệ che phủ của xe máy trong các vùng quan tâm. Trên cơ sở đó, số lượng xe máy trong từng vùng được ước lượng, góp phần nâng cao độ chính xác tổng thể của quá trình đếm phương tiện.

- **Khối dự báo lưu lượng và điều chỉnh tín hiệu đèn giao thông:** None
- **Khối mô phỏng:** None
- **Khối hiển thị và đánh giá kết quả:** None

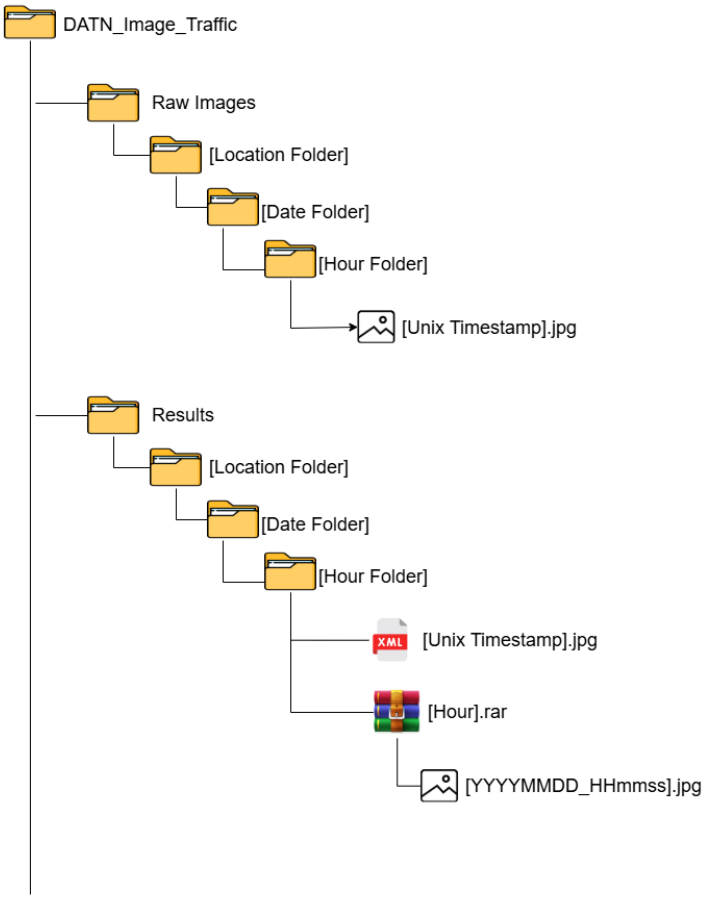
3.2.2 Thiết kế khối thu thập dữ liệu

None

3.2.3 Thiết kế khối lưu trữ

Khối lưu trữ dữ liệu đóng vai trò tiếp nhận, tổ chức và quản lý toàn bộ dữ liệu trong hệ thống, bao gồm ảnh snapshot thu thập từ camera giao thông, dữ liệu sau tiền xử lý, kết quả nhận diện - đếm phương tiện, cũng như các tập dữ liệu đầu vào/đầu ra phục vụ mô phỏng và dự báo. Trong đề tài này, Google Drive được sử dụng như một nền tảng lưu trữ đám mây, đáp ứng yêu cầu về dung lượng, khả năng truy cập linh hoạt và thuận tiện cho việc chia sẻ, sao lưu dữ liệu trong quá trình nghiên cứu.

Việc sử dụng Google Drive cho phép hệ thống tách biệt hoàn toàn giữa khâu xử lý và khâu lưu trữ, từ đó giảm phụ thuộc vào hạ tầng máy chủ cục bộ và tăng tính linh hoạt trong quá trình mở rộng hoặc thay đổi môi trường triển khai. Khối lưu trữ không thực hiện các chức năng xử lý dữ liệu phức tạp mà chỉ đảm nhiệm vai trò lưu trữ tệp và cung cấp dữ liệu cho các khối xử lý tiếp theo thông qua các script truy xuất tự động.



Hình 3.2.2: Cấu trúc lưu trữ dữ liệu trên Google Drive

Như trên hình **??**, dữ liệu được tổ chức theo mô hình cây phân cấp, trong đó:

- **DATN_Image_Traffic:** Thư mục chủ đạo chứa dữ liệu đầu vào thô và dữ liệu đã qua xử lý, chia làm hai nhánh chính:
  - **Raw Images:** Dữ liệu đầu vào thu thập trực tiếp từ camera giao thông (giaothong.hochiminhcity.gov.vn). Cấu trúc gồm:
    - **Location Folder:** Tên thư mục theo địa điểm thực tế (Ví dụ: `Nga_Tu_So`, `Cau_Giay`).
    - **Date Folder:** Phân loại theo ngày thu thập (Định dạng: `YYYY-MM-DD`).
    - **Hour Folder:** Phân loại theo khung giờ (Ví dụ: `01h`, `02h`...).
    - **[Unix_Timestamp].jpg:** Các tệp ảnh gốc với tên là Unix Timestamp để đảm bảo tính duy nhất và tự động sắp xếp theo thời gian.

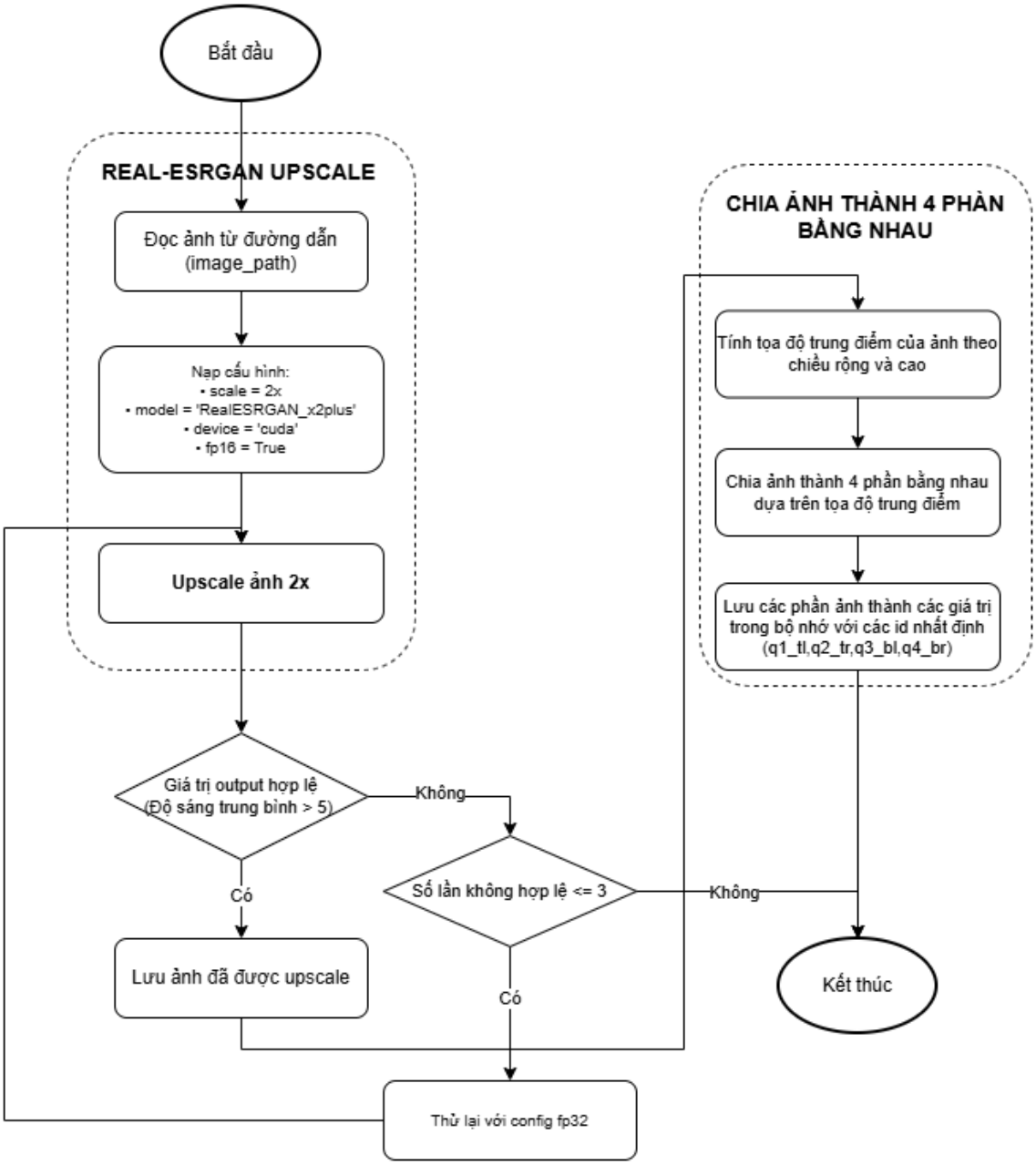
- **Results:** Lưu trữ kết quả sau khi xử lý/nhận diện, duy trì cấu trúc tương tự nhánh Raw Images để dễ dàng đối chiếu:
  - **[Unix_Timestamp].csv:** Tệp tin chứa nhãn (labels) hoặc metadata tương ứng với ảnh gốc.
  - **[Hour].rar:** Tệp nén toàn bộ dữ liệu của khung giờ đó để tối ưu lưu trữ.
  - **[YYYYMMDD_HHmmss].jpg:** Ảnh kết quả (đã vẽ bounding box...) đặt tên theo định dạng thời gian truyền thống để người dùng dễ đọc bằng mắt thường.

3.2.4 Thiết kế khối tiền xử lý ảnh

Khối tiền xử lý ảnh có vai trò chuẩn hóa và nâng cao chất lượng đầu vào trước khi đưa vào khối nhận diện và phân tích phương tiện. Hệ thống thực hiện các bước tiền xử lý chính sau:

1. **Tăng cường độ phân giải và chuẩn hóa kích thước ảnh:** Sử dụng Real-ESRGAN để nâng cao độ phân giải và giảm nhiễu cho các ảnh snapshot, ngoài ra còn tăng kích thước ảnh về chuẩn cố định nhằm đảm bảo tính nhất quán và cải thiện khả năng nhận diện của mô hình học sâu.

2. **Chia đều ảnh thành các phần:** Ảnh sau khi được nâng cao chất lượng được chia thành bốn vùng con có kích thước bằng nhau. Việc chia nhỏ ảnh giúp mô hình tập trung tốt hơn vào các đặc trưng cục bộ, từ đó nâng cao độ chính xác trong việc phát hiện phương tiện, đặc biệt trong các tình huống mật độ giao thông cao hoặc khi các đối tượng bị che khuất một phần.



Hình 3.2.3: Lưu đồ chi tiết quá trình tiền xử lý ảnh

Giải thích lưu đồ tiền xử lý ảnh ở hình **??**:

Ảnh đầu vào trước hết được đưa vào mô hình Real-ESRGAN đã được huấn luyện sẵn (pre-trained model) với các tham số cấu hình gồm hệ số phóng đại $scale = 2$, mô hình *RealESRGAN_x2plus*, thiết bị xử lý *cuda* và chế độ tính toán bán chính xác *fp16*.

Sau khi quá trình nâng cao độ phân giải hoàn tất, chất lượng ảnh đầu ra được đánh giá thông qua việc tính toán giá trị độ sáng trung bình của các pixel trong ảnh. Nếu giá

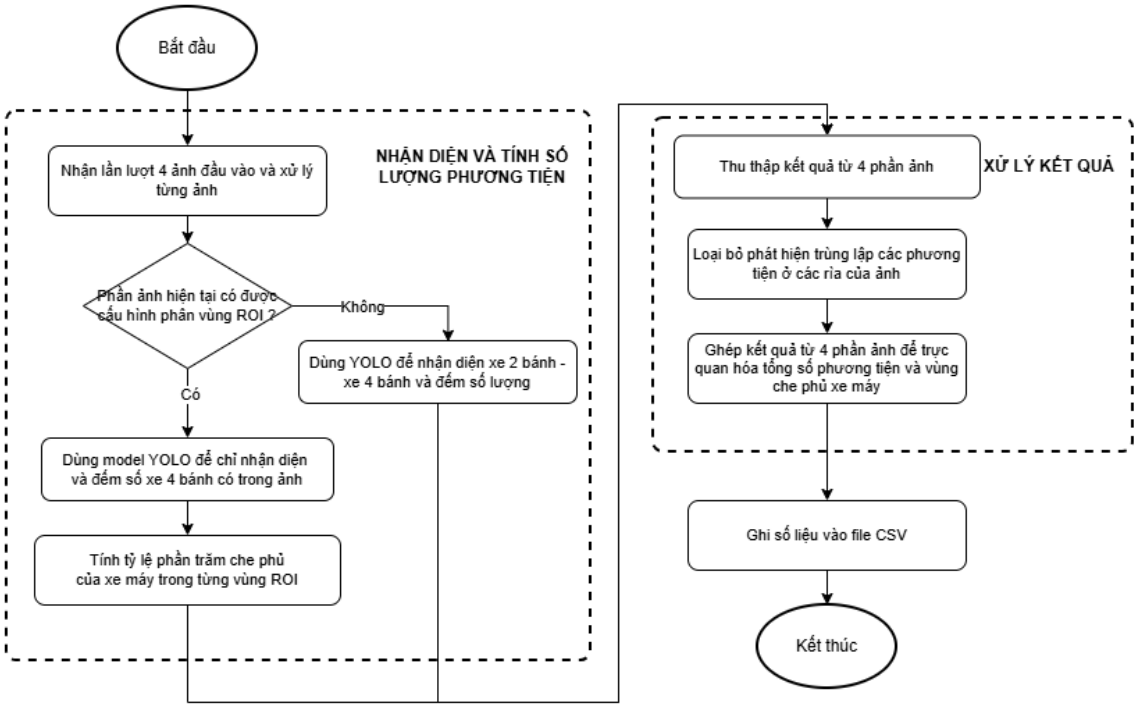
trị này nhỏ hơn một ngưỡng xác định (cụ thể là 5), ảnh được xem là không hợp lệ do quá tối và có khả năng phát sinh lỗi trong quá trình upscale. Trong trường hợp này, hệ thống tiến hành thực hiện lại bước nâng cao độ phân giải với cấu hình tương tự, nhưng chuyển sang chế độ tính toán toàn chính xác (*fp32*) nhằm tăng độ ổn định và độ chính xác của kết quả.

Ngược lại, nếu ảnh đầu ra đạt yêu cầu về độ sáng, ảnh sẽ được chia thành bốn vùng con có kích thước bằng nhau. Quá trình này được thực hiện bằng cách xác định tọa độ trung điểm theo chiều rộng và chiều cao của ảnh, từ đó cắt ảnh thành bốn phần tương ứng với các góc *Top-Left*, *Top-Right*, *Bottom-Left* và *Bottom-Right*.

Cuối cùng, các ảnh con sau khi chia không được lưu dưới dạng các tệp ảnh riêng lẻ, mà được lưu trữ dưới dạng các giá trị trong bộ nhớ cùng với các id để nhận biết được các phần ảnh để phục vụ cho quá trình nhận diện và phân tích phuwogn tiện ở khối tiếp theo.

3.2.5 Thiết kế khối nhận diện và phân tích phương tiện

Khối nhận diện và phân tích phương tiện sử dụng mô hình học sâu YOLOv11 đã được huấn luyện trước trên dataset COCO (pre-trained model) để phát hiện và phân loại các phương tiện giao thông trong từng ảnh snapshot. Tuy nhiên, bởi vì dữ liệu đầu vào có chất lượng không tốt, góc quay không tối ưu và mật độ giao thông cao, việc nhận diện chính xác các phương tiện, đặc biệt là xe máy trở nên khó khăn. Do đó, ngoài việc sử dụng mô hình YOLOv11 để phát hiện và phân loại phương tiện thì cần có thêm các bước xử lý ảnh bổ sung nhằm nâng nâng cao độ chính xác trong việc đếm xe máy, tính toán tỷ lệ che phủ của xe máy trong các vùng quan tâm (ROI - Region of Interest).



Hình 3.2.4: Lưu đồ quá trình nhận diện và đếm phương tiện

Giải thích lưu đồ nhận diện và đếm phương tiện ở hình **??**:

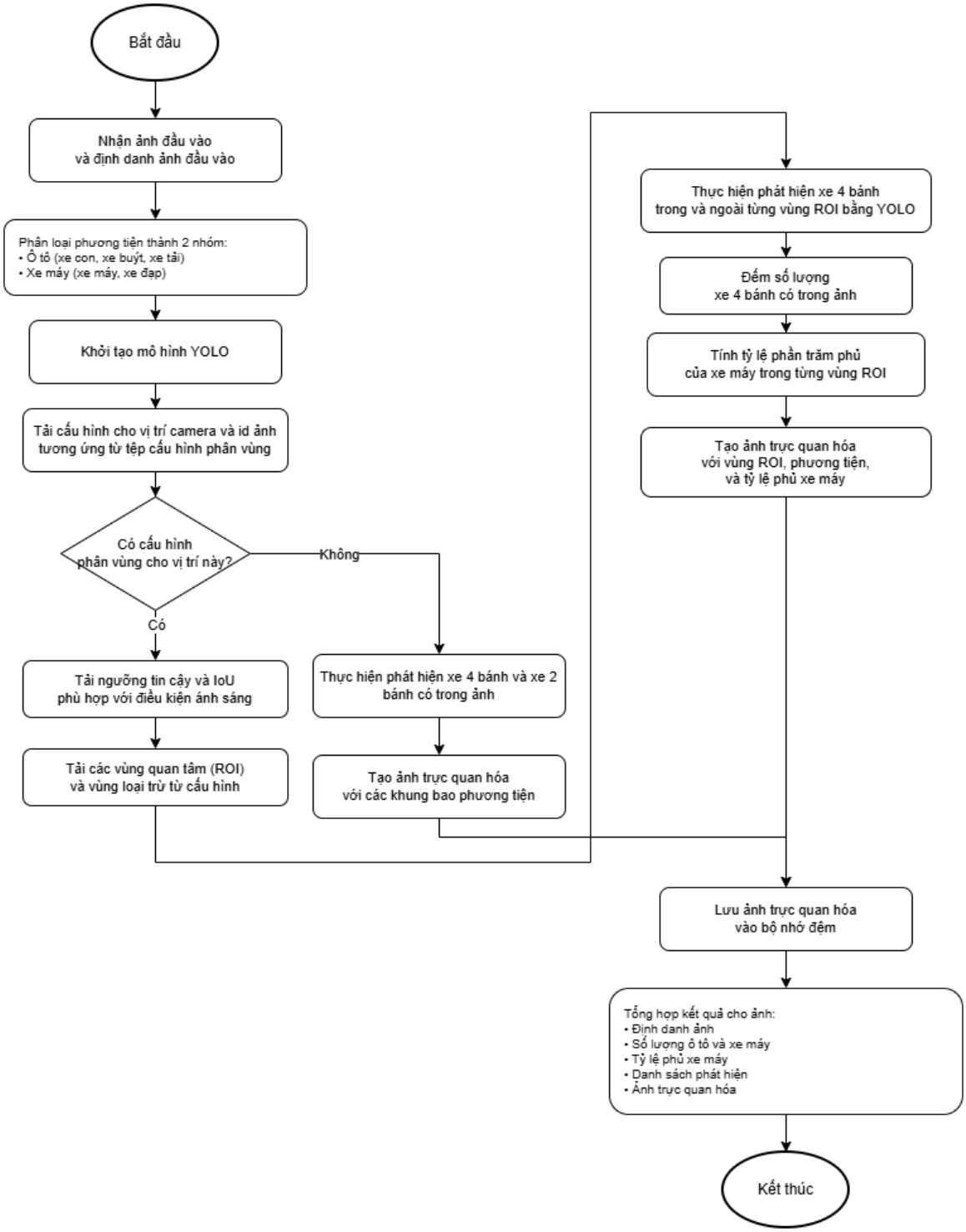
Khối này bắt đầu bằng việc tiếp nhận chính xác bốn ảnh con đã được tiền xử lý từ khối trước và tiến hành xử lý lần lượt từng ảnh. Trước hết, hệ thống kiểm tra xem ảnh con đầu vào có thuộc danh sách các Tile ID được cấu hình phân vùng ROI (Region of Interest) hay không. Nếu ảnh không nằm trong danh sách này, ảnh sẽ được đưa trực tiếp vào mô hình YOLOv11 để thực hiện phát hiện và phân loại các phương tiện, bao gồm xe hai bánh và xe bốn bánh. Ngược lại, nếu ảnh con thuộc một Tile ID có cấu hình phân vùng ROI, ảnh sẽ được xử lý bổ sung bằng các thuật toán xử lý ảnh nhằm ước lượng số lượng xe máy trong vùng quan tâm, đồng thời vẫn thực hiện nhận diện phương tiện bằng mô hình YOLOv11. Sau khi hoàn tất quá trình xử lý đối với cả bốn ảnh con, các kết quả đếm phương tiện theo từng phần ảnh được tổng hợp lại. Hệ thống tiếp tục kiểm tra sự trùng lặp của các phương tiện xuất hiện tại vùng rìa giữa các ảnh con và loại bỏ các trường hợp bị đếm trùng. Cuối cùng, kết quả tổng hợp được lưu trữ dưới dạng tệp csv. Đồng thời, các ảnh con được ghép lại để tạo thành ảnh kết quả, trong đó các phương tiện được phát hiện được biểu diễn bằng các bounding box, kèm theo phần che phủ của xe máy trong các vùng phân đoạn (nếu có).

Đối với việc chọn vùng quan tâm (ROI) để ước lượng số lượng xe máy thì trong đề tài này chỉ lựa chọn các vùng có mật độ xe máy cao và thường bị che khuất trong ảnh snapshot gốc. Các vùng phân đoạn được xác định thông qua một công cụ hỗ trợ được xây dựng riêng bằng Python. Công cụ này cho phép người vận hành khoanh vùng các khu vực cần khai thác trên ảnh,sau đó xuất ra tọa độ các điểm biên của vùng ROI theo hệ tọa độ pixel của ảnh. Các tọa độ ROI được biểu diễn dưới dạng các cặp điểm (x, y) trong hệ tọa độ ảnh, tuân theo quy ước của thư viện OpenCV, với gốc tọa độ nằm tại góc trên bên trái ảnh. Những thông tin này được lưu trữ trong tệp cấu hình YAML tương ứng với từng địa điểm và được hệ thống sử dụng trong quá trình xử lý và phân tích ảnh như hình **??**.

```yaml
locations:
 - name: "Dinh Tien Hoang - Vo Thi Sau 2"
   segments:
     - id: "q1_tl"
       yolo:
           confidence_threshold: 0.3
           iou_threshold: 0.45
           actual_conf_threshold: 0.1
       roi_points:
         - roi_point: [[94, 149], [180, 123], [371, 286], [103, 287], [146, 259], [142, 238], [100, 206], [131, 197]]
         - roi_point: [[187, 145], [287, 131], [507, 231], [508, 282], [383, 283]]
       exclusion_zones:
         - roi_point: [[153, 144], [161, 133], [367, 286], [316, 286]]
```

Hình 3.2.5: Cấu trúc file YAML cấu hình vùng quan tâm (ROI)

Ngoài ra đối với các vùng có vật cản hoặc có yếu tố gây nhiễu trong vùng segment như bóng cây, biển quảng cáo, cột đèn... thì trong đề tài này còn cấu hình thêm các ROI để loại bỏ các yếu tố nhiễu này nhằm tránh ảnh hưởng đến kết quả đếm phương tiện như *exclusion_zones* trên hình **??**.



Hình 3.2.6: Lưu đồ xử lý từng ảnh con

Giải thích lưu đồ xử lý từng ảnh con ở hình **??**:

Trước hết, hệ thống nạp cấu hình phân vùng ROI (Region of Interest) từ tệp YAML đã được trình bày ở phần trước. Tiếp theo, mỗi ảnh con đầu vào được kiểm tra xem ID

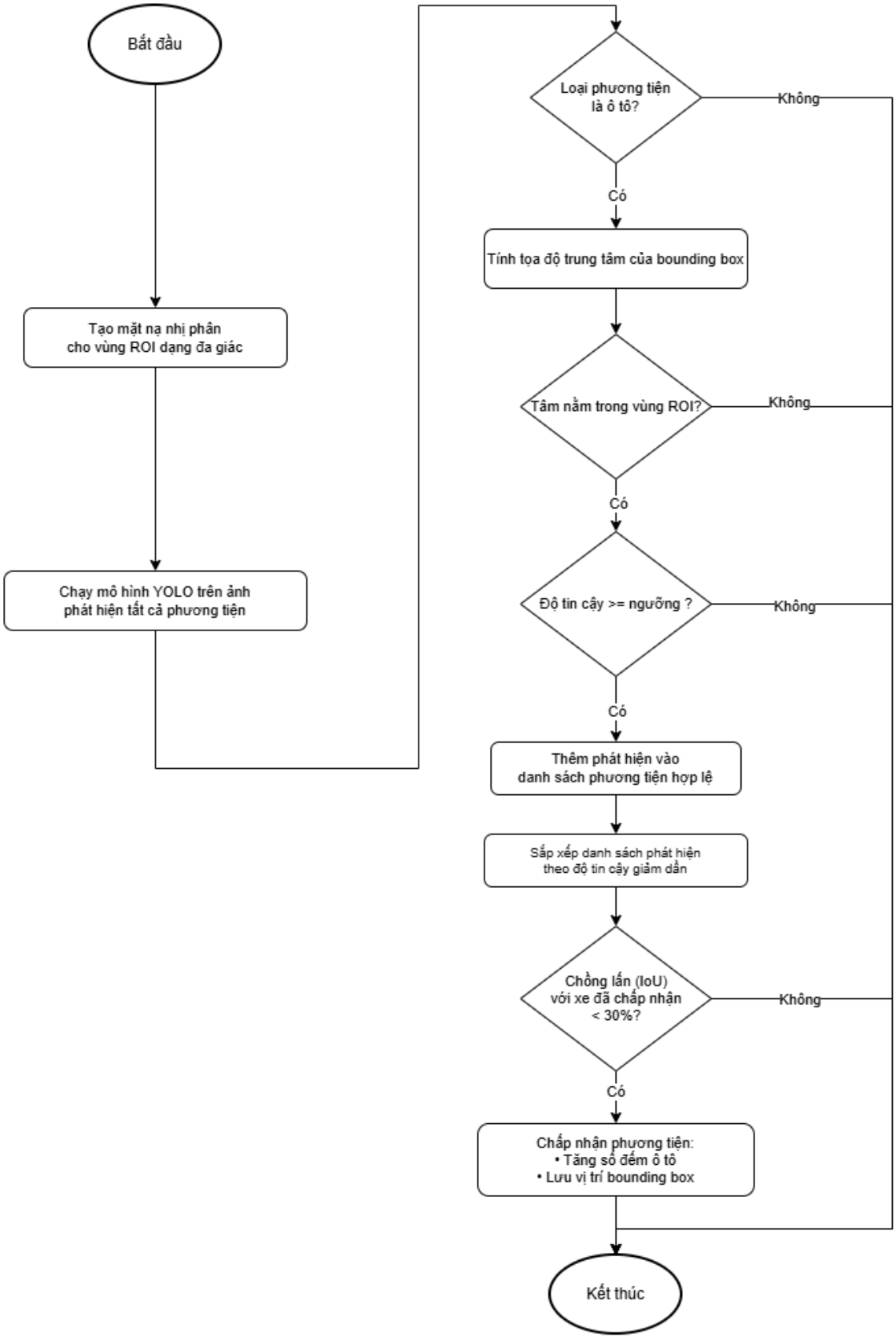
của phần ảnh đó có nằm trong danh sách các vùng được cấu hình ROI hay không.

Trong trường hợp ảnh con không thuộc danh sách ROI, ảnh sẽ được đưa trực tiếp vào mô hình YOLOv11 để thực hiện phát hiện và phân loại phương tiện. Ngược lại, nếu ảnh con thuộc vùng có cấu hình ROI, các tham số tương ứng như ngưỡng tin cậy (confidence threshold), ngưỡng IOU và tọa độ vùng quan tâm sẽ được trích xuất từ cấu hình để phục vụ cho các bước xử lý tiếp theo.

Tại giai đoạn này, mô hình YOLOv11 chỉ được sử dụng để phát hiện phương tiện bốn bánh. Đối với phương tiện hai bánh, hệ thống không áp dụng trực tiếp phương pháp phát hiện dựa trên bounding box do đặc thù kích thước nhỏ, mật độ cao và hiện tượng che khuất phổ biến, dẫn đến độ chính xác thấp, số lượng phát hiện không ổn định và kết quả trực quan không phản ánh đúng mật độ thực tế trong vùng quan tâm.

Thay vào đó, số lượng xe hai bánh được ước lượng gián tiếp thông qua tỷ lệ che phủ trong vùng ROI. Cụ thể, hệ thống áp dụng một số kỹ thuật xử lý ảnh bao gồm: Polygon Fill, Rectangle Fill, các phép toán logic theo pixel và đếm số pixel khác không (Non-zero Pixel Count) nhằm xác định tỷ lệ diện tích bị che phủ bởi xe máy trong vùng quan tâm. Trên cơ sở tỷ lệ che phủ này, số lượng xe hai bánh được ước lượng tương ứng.

Cuối cùng, kết quả đếm phương tiện hai bánh và bốn bánh, tỷ lệ che phủ của xe máy, cùng với ảnh trực quan hóa được lưu trữ trong bộ nhớ sẽ được tổng hợp và trả về cho khối xử lý chính của hệ thống.



Hình 3.2.7: Lưu đồ nhận diện phương tiện 4 bánh có trong vùng ROI

Giải thích lưu đồ nhận diện phương tiện 4 bánh có trong vùng ROI ở hình **??**:

Trước hết, hệ thống tạo một mặt nạ nhị phân cho vùng ROI dạng đa giác dựa trên các tọa độ được trích xuất từ tệp cấu hình YAML, nhằm xác định khu vực cần phân tích trong ảnh. Tiếp theo, mô hình YOLOv11 được sử dụng để phát hiện các phương tiện trong ảnh, với các tham số ngưỡng tin cậy và ngưỡng IOU được cấu hình trước.

Sau khi thu được kết quả phát hiện, hệ thống lần lượt duyệt qua từng đối tượng và chỉ xem xét các phương tiện thuộc lớp xe bốn bánh. Đối với mỗi phương tiện hợp lệ về mặt lớp, hệ thống tính toán tọa độ bounding box và xác định xem tâm của bounding box có nằm trong vùng ROI hay không. Các phương tiện nằm ngoài vùng quan tâm sẽ không được đưa vào các bước xử lý tiếp theo.

Đối với các phương tiện bốn bánh có tâm bounding box nằm trong vùng ROI, hệ thống tiếp tục kiểm tra độ tin cậy của dự đoán. Những đối tượng có độ tin cậy nhỏ hơn ngưỡng cấu hình sẽ bị loại bỏ. Các phương tiện thỏa mãn đầy đủ các điều kiện trên sẽ được đưa vào danh sách phương tiện hợp lệ. Danh sách này sau đó được sắp xếp theo độ tin cậy giảm dần nhằm ưu tiên giữ lại các phát hiện có độ tin cậy cao trong quá trình xử lý trùng lặp.

Tiếp theo, hệ thống thực hiện loại bỏ các phát hiện trùng lặp dựa trên ngưỡng IOU đã được cấu hình. Trong trường hợp hai bounding box có giá trị IOU vượt quá ngưỡng cho phép, đối tượng có độ tin cậy thấp hơn sẽ bị loại bỏ khỏi danh sách.

Cuối cùng, số lượng phương tiện bốn bánh hợp lệ còn lại được cộng dồn vào biến đếm tổng. Đồng thời, thông tin vị trí của các bounding box này được lưu lại để phục vụ cho việc tính toán tỷ lệ che phủ của xe hai bánh cũng như trực quan hóa kết quả trong các bước xử lý tiếp theo.

TÀI LIỆU THAM KHẢO