

HỌC VIỆN KỸ THUẬT QUÂN SỰ

HỌ VÀ TÊN: HÀ HUY GIANG

KHÓA: 50

HỆ ĐÀO TẠO: DÀI HẠN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

XÂY DỰNG HỆ THỐNG GIÁM SÁT XE Ô TÔ TRÊN ĐIỆN
RỘNG TRÊN CÔNG NGHỆ NHẬN DẠNG BIỂN SỐ XE

NĂM 2019

HỌC VIỆN KỸ THUẬT QUÂN SỰ

HỌ VÀ TÊN: HÀ HUY GIANG

KHÓA: 50

HỆ ĐÀO TẠO: DÀI HẠN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

MÃ SỐ: 7480201

XÂY DỰNG HỆ THỐNG GIÁM SÁT XE Ô TÔ TRÊN ĐIỆN
RỘNG TRÊN CÔNG NGHỆ NHẬN DẠNG BIỂN SỐ XE

Cán bộ hướng dẫn khoa học: Trung tá, GV. TS. Nguyễn Quốc Khánh

NĂM 2019

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên: HÀ HUY GIANG

Lớp: Tin học

Khóa: 50

Ngành: Công nghệ thông tin

Chuyên ngành: Công nghệ thông tin

1. Tên đề tài:

Xây dựng hệ thống giám sát xe ô tô trên diện rộng trên công nghệ nhận dạng biển số xe

2. Các số liệu ban đầu:

- Quyết định Giao đồ án tốt nghiệp đại học – Học viện KTQS
- Tài liệu tham khảo

3. Nội dung bản thuyết minh:

- Mở đầu
- Chương 1: Khảo sát hệ thống
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phân tích và thiết kế hệ thống
- Chương 4: Xây dựng hệ thống
- Chương 5: Cài đặt và triển khai
- Kết luận
- Danh mục tài liệu tham khảo

4. Số lượng, nội dung các bản và các sản phẩm cụ thể:

- Được sử dụng máy tính và máy chiếu để trình chiếu.

5. Cán bộ hướng dẫn:

- Họ và tên: TS. Nguyễn Quốc Khánh
- Cấp bậc: Trung tá
- Chức vụ: Phó chủ nhiệm bộ môn
- Đơn vị: Bộ môn Công nghệ phần mềm- Khoa Công nghệ thông tin
- Nội dung hướng dẫn: Toàn bộ đồ án

Ngày giao: 24/06/2019

Ngày hoàn thành: 30/10/2019

Hà Nội, ngày thángnăm 2019

Chủ nhiệm bộ môn

Cán bộ hướng dẫn

2// TS. Nguyễn Quốc Khánh

Học viên thực hiện

Đã hoàn thành và nộp đồ án ngày 30 tháng 10 năm 2019

Hà Huy Giang

KÝ HIỆU VIẾT TẮT

STT	Từ viết tắt	Từ tiếng Anh	Từ tiếng Việt
1	ALPR	Automatic License Plate Recognition	Nhận diện biển số xe tự động
2	OCR	Optical Character Recognition	Nhận diện ký tự quang học
3	CNN	Convolutional Neural Network	Mạng nơ ron tích chập
4	SSD	Single Shot MultiBox Detector	Nhận diện nhiều hộp một lần
5	SSPNet	Spatial Pyramid Pooling Network	Mạng tổng hợp kim tự tháp không gian
6	FPS	Frame per second	Khung hình trên giây
7	STR	Scene Text Recognition	Nhận diện văn bản hình ảnh
8	CSDL		Cơ sở dữ liệu

DANH MỤC HÌNH ẢNH

Hình 1.1: Giao diện phần mềm giám sát lỗi vi phạm giao thông	3
Hình 1.2. Hệ thống nhận dạng biển số xe tự động	4
Hình 2.1. Mối quan hệ giữa các phương pháp học máy	9
Hình 2.2. Sự gia tăng về số lượng các ấn phẩm liên quan đến phát hiện đối tượng từ 1998 đến 2018	13
Hình 2.3. Sự phát triển của các thuật toán phát hiện đối tượng	13
Hình 2.4. Mô hình SSD	18
Hình 2.5. So sánh giữa hai mô hình SSD và YOLO	20
Hình 2.6. Các bộ lọc tích chập tiêu chuẩn	22
Hình 2.7. Main building block MobileNet V2	26
Hình 2.8. Đầu vào và đầu ra của khối là các tensor chiều thấp	27
Hình 2.9. Mô hình kiến trúc SSD MobileNet	28
Hình 2.10. Quá trình nhận diện text từ ảnh chụp	29
Hình 2.11. Mô hình chung của Gmap.NET	36
Hình 3.1. Mô hình use case cho vai trò người quản lý hệ thống	42
Hình 3.2. Mô hình use case cho vai trò nhân viên giám sát hệ thống	43
Hình 3.3. Sơ đồ cơ sở dữ liệu logic	50
Hình 3.4. Giao diện đăng nhập	55
Hình 3.5. Giao diện xem camera dạng lưới	55
Hình 3.6. Giao diện bản đồ camera	56
Hình 3.7. Giao diện danh sách camera	57
Hình 3.8. Giao diện thêm, cập nhật camera	57
Hình 3.9. Giao diện video playback	58

Hình 3.10. Giao diện xem camera đơn	59
Hình 3.11. Giao diện theo dõi hành trình	59
Hình 3.12. Giao diện tra cứu biển số	60
Hình 3.13. Giao diện kết quả nhận diện biển số theo camera	61
Hình 3.14. Giao diện quản lý người dùng	61
Hình 3.15. Giao diện quản lý vai trò	62
Hình 4.1. Mô hình hệ thống.....	64
Hình 4.2. Huấn luyện mô hình nhận diện biển số xe.....	65
Hình 4.3. Sơ đồ nhận diện biển số xe	66
Hình 4.4. Biển số xe một dòng và hai dòng qua bước phát hiện.....	66
Hình 4.5. Huấn luyện mô hình nhận diện ký tự.....	67
Hình 4.6. Sơ đồ nhận diện ký tự	67
Hình 4.7. Các điểm Harris Corner	68
Hình 4.8. Ảnh biển số xe sau xử lý ảnh.....	68
Hình 5.1. Server chính quản lý và xử lý dữ liệu.....	72
Hình 5.2. Module nhận diện biển số xe	72
Hình 5.3. Chọn vị trí camera trên bản đồ	73
Hình 5.4. Xem vị trí và trạng thái của các camera	74
Hình 5.5. Hành trình của phương tiện trên bản đồ	74
Hình 5.6. Menu chức năng camera	75
Hình 5.7. Menu chức năng theo dõi phương tiện	75
Hình 5.8. Menu chức năng người dùng	75

DANH MỤC BẢNG BIỂU

Bảng 2.1. Kiến trúc MobileNet.....	24
Bảng 2.2. Tài nguyên cho mỗi loại tầng.....	25
Bảng 3.1. Camera Type	50
Bảng 3.2. Camera.....	50
Bảng 3.3. Resolution.....	51
Bảng 3.4. Camera_Resolution	51
Bảng 3.5. Video Playback.....	52
Bảng 3.6. License Plate.....	52
Bảng 3.7. License Plate Camera	53
Bảng 3.8. User Monitoring Plate	53
Bảng 3.9. User.....	54
Bảng 3.10. Role.....	54
Bảng 5.1. Cấu hình máy triển khai hệ thống	71

MỤC LỤC

MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Mục đích đề tài	1
3. Đối tượng nghiên cứu.....	1
Chương 1. KHẢO SÁT HỆ THỐNG	2
1.1. Tổng quan về tình hình giám sát giao thông trên diện rộng.....	2
1.2 Khái niệm về nhận dạng biển số xe	4
1.2.1 Khái niệm.....	4
1.2.3. Ứng dụng của hệ thống nhận dạng biển số xe:	6
1.3. Các kỹ thuật nhận dạng biển số xe:	6
1.3.1. Nhận dạng biển số xe sử dụng kỹ thuật xử lý ảnh.....	7
1.3.2. Nhận dạng biển số xe sử dụng kỹ thuật học máy.	7
Kết luận chương 1	8
Chương 2. CƠ SỞ LÝ THUYẾT.....	9
2.1. Tổng quan về học máy	9
2.1.1. Tổng quan	9
2.1.2. Học có giám sát (Supervised Learning).....	10
2.1.3. Học không giám sát (Unsupervised Learning)	11
2.1.4. Học bán giám sát (Semi-supervised learning)	11
2.2. Bài toán phát hiện đối tượng.....	12
2.2.1. Cách nhận diện truyền thống:	13
2.2.2. Nhận diện 2 giai đoạn dựa trên CNN.....	14

2.2.3 Nhận diện một giai đoạn dựa trên CNN	16
2.3. Mô hình SSD Mobilenet	18
2.3.1 SSD (Single Shot MultiBox Detector).....	18
2.3.2. MobileNet	21
2.3.3. SSD MobileNet	28
2.4. Nhận dạng văn bản (Scene Text Recognition)	28
2.4.1. Giai đoạn chuyển đổi	30
2.4.2. Giai đoạn trích xuất đặc trưng	30
2.4.3. Giai đoạn mô hình hóa trình tự	30
2.4.4. Giai đoạn dự đoán	31
2.5. Bản đồ số	31
2.5.1. Khái niệm về bản đồ số.....	31
2.5.2. Google Map API	32
2.5.3. Direction API	32
2.5.4. Nền tảng GMap.Net	35
2.6. Hướng tiếp cận của đề án.....	36
Kết luận chương 2	37
Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	38
3.1. Các thành phần, chức năng chính và các tác nhân của hệ thống..	38
3.1.1 Các thành phần và chức năng chính	38
3.1.2. Các tác nhân của hệ thống:	39
3.2. Cấu trúc hệ thống:	39
3.2.1. Phân hệ quản lý camera	39
3.2.2. Phân hệ theo dõi phương tiện	40

3.2.3. Phân hệ quản lý người dùng:	41
3.3. Biểu đồ Use Case	41
3.3.1. Các User Case của hệ thống.....	41
3.3.2. Đặc tả use case trong hệ thống.....	43
3.4. Thiết kế kiến trúc	49
3.5. Thiết kế dữ liệu.....	49
3.5.1. Sơ đồ logic	49
3.5.2. Cơ sở dữ liệu vật lý	50
3.6. Thiết kế giao diện	54
3.6.1. Giao diện đăng nhập	54
3.6.2. Giao diện xem camera dạng lưới	55
3.6.3. Giao diện bản đồ camera.....	56
3.6.4. Giao diện danh sách camera	56
3.6.5. Giao diện thêm và cập nhật thông tin camera.....	57
3.6.6. Giao diện videoplayback.....	58
3.6.7. Giao diện xem camera đơn	58
3.6.8. Giao diện theo dõi phương tiện.....	59
3.6.9. Giao diện xem dữ liệu nhận diện biển số xe của một biển số xe nhất định.....	60
3.6.10. Giao diện xe kết quả nhận dạng biển số xe theo camera	60
3.6.11. Giao diện quản lý người dùng.....	61
3.6.12. Giao diện quản lý vai trò.....	62
Kết luận chương 3	63
Chương 4. XÂY DỰNG HỆ THỐNG	64

4.1. Mô hình hệ thống	64
4.2. Xây dựng các module.....	64
4.2.1. Module nhận diện biển số xe	64
4.2.2. Server chính quản lý và xử lý dữ liệu	68
4.2.3. Phần mềm phía client.....	68
Kết luận chương 4	69
Chương 5. CÀI ĐẶT VÀ TRIỂN KHAI.....	70
5.1. Yêu cầu.....	70
5.1.1. Yêu cầu phần cứng.....	70
5.1.2. Yêu cầu phần mềm.....	70
5.2. Cài đặt hệ thống	70
5.2.1. Phần cứng.....	70
5.2.2. Môi trường cài đặt.....	71
5.3. Thực thi phần mềm.....	71
5.3.1. Server chính quản lý và xử lý dữ liệu	71
5.3.2. Module nhận diện biển số xe	72
5.3.3. Phần mềm phía client.....	73
Kết luận chương 5	76
KẾT LUẬN	77
Các kết quả đã đạt được.....	77
Những điểm còn hạn chế của đồ án.....	77
Hướng phát triển của đồ án	77
TÀI LIỆU THAM KHẢO	79

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay trên thế giới bên cạnh việc tăng trưởng kinh tế là sự phát triển của các ngành khoa học kỹ thuật nói chung, mà trong đó ngành công nghiệp sản xuất các phương tiện giao thông là một trong những ngành có tốc độ phát triển cực nhanh. Sự phát triển ấy được thể hiện rõ nhất thông qua những năm gần đây, các phương tiện giao thông ngày một tăng cao và đa dạng. Tuy nhiên điều đó lại gây ra áp lực đối với những người và cơ quan các cấp quản lý, làm cho công tác quản lý và giám sát sẽ khó khăn hơn...

Và đây cũng là một trong những vấn nạn ở Việt Nam. Công tác quản lý phương tiện giao thông nói chung và quản lý ô tô, xe máy là vô cùng phức tạp, cũng như nhận diện, xử phạt các hành vi vi phạm giao thông, sẽ tốn nhiều thời gian và công sức hơn.

Để giảm nhân lực và xử lý trong việc công tác quản lý, kiểm soát phương tiện giao thông, trên thế giới đã nhanh chóng xây dựng hệ thống giám sát tự động đối với các phương tiện giao thông lấy biển số xe là mục tiêu giám sát. Hệ thống này đã được sử dụng rộng rãi tuy nhiên ở Việt Nam đây vẫn là một lĩnh vực mới mẻ .

Xuất phát từ lý do trên, tôi chọn đề tài: “Xây dựng hệ thống giám sát xe ô tô trên diện rộng dựa trên công nghệ nhận dạng biển số xe”.

2. Mục đích đề tài

- Tìm hiểu về các công nghệ nhận dạng biển số xe, áp dụng vào xây dựng hệ thống giám sát xe ô tô trên diện rộng.

3. Đối tượng nghiên cứu

- Các công nghệ nhận dạng biển số xe
- Các loại camera hiện có.

Chương 1

KHẢO SÁT HỆ THỐNG

1.1. Tổng quan về tình hình giám sát giao thông trên diện rộng

Tại Hà Nội, riêng Phòng Cảnh sát giao thông Hà Nội đang quản lý, sử dụng gần 600 camera các loại, trong đó có hơn 300 camera để theo dõi lưu lượng phương tiện, hơn 100 camera phục vụ xử phạt vi phạm giao thông và gần 100 thiết bị giám sát giao thông. Ngoài ra, tại nhiều nút giao thông còn có camera của công an các quận; camera của VOV giao thông; camera theo dõi điểm ngập úng của ngành thoát nước... Nhờ hệ thống này, tình trạng người tham gia giao thông vi phạm các lỗi như lấn làn, vượt đèn đỏ, dừng đỗ không đúng vị trí... đã giảm đáng kể. Hà Nội đang khẩn trương nghiên cứu tích hợp không chỉ camera giám sát của các cơ quan nhà nước, mà còn của người dân, doanh nghiệp (sân bay, nhà ga, bến xe, các cơ quan hành chính, ngõ phố...) để hỗ trợ công tác điều hành giao thông chung..

Tại TP Hồ Chí Minh, trung tâm giám sát và điều khiển giao thông giai đoạn 1 (thuộc dự án Nâng cấp hệ thống điều khiển giao thông hiện hữu) đã hoàn thành, tổng mức đầu tư khoảng 250 tỷ đồng, Trung tâm có nhiệm vụ kiểm soát giao thông trên các tuyến đường chính, nút giao thông khu vực trung tâm thành phố, với 4 chức năng: điều khiển đèn tín hiệu; giám sát và cung cấp thông tin giao thông; hỗ trợ xử lý vi phạm. Trung tâm giám sát và điều khiển giao thông vận hành 24/7 trong cả 365 ngày có thể được xem như "mắt thần". Thông tin sẽ lập tức được chia sẻ trên cổng thông tin giao thông cho người dân cập nhật, tránh các điểm kẹt xe, chống ùn tắc, xử lý các sự cố hạ tầng hữu hiệu. Ở giai đoạn 2, sau năm 2020 sẽ hoàn thành Trung tâm điều hành giao thông thông minh có quy mô toàn thành phố, tổng vốn đầu tư 250 triệu USD. Trung tâm điều hành giao thông thông minh sẽ quản lý trực tiếp toàn bộ hệ thống điều khiển giao thông của thành phố cũng như cung cấp thông tin dữ liệu, phục vụ cho việc xử lý các vi phạm;

giám sát, đảm bảo an ninh trật tự đô thị; xử lý các sự cố khẩn cấp; phòng chống cháy nổ, ngập lụt; quản lý bến bãi.

STT	Ảnh biển số	Biển số xe	Lỗi	Thời gian	Vị trí	Camera
21		75A12980		14/06/2019 18:19	Nga 6 LTK+HN	HUE_CAM001
22		75-B1 131.00		14/06/2019 18:09	Nga 6 LTK+HN	HUE_CAM001
23		75-L2 237.3		14/06/2019 18:09	Nga 6 LTK+HN	HUE_CAM001
24		75-B1 204.26		14/06/2019 18:02	Nga 6 LTK+HN	HUE_CAM001
25		75-L1 028.02		14/06/2019 17:45	Nga 6 LTK+HN	HUE_CAM001
26		75-B1 248.65		14/06/2019 17:37	Nga 6 LTK+HN	HUE_CAM001
27		75-H7 262.6		14/06/2019 17:25	Nga 6 LTK+HN	HUE_CAM001
28		75-L1 680.9		14/06/2019 17:20	Nga 6 LTK+HN	HUE_CAM001
29		75-B1 277.51		14/06/2019 17:16	Nga 6 LTK+HN	HUE_CAM001
30		75-F1 281.47		14/06/2019 17:08	Nga 6 LTK+HN	HUE_CAM001

© 2017 Bản quyền thuộc Tập đoàn Viễn thông Quân đội Viettel

Hình 1.1: Giao diện phần mềm giám sát lỗi vi phạm giao thông

Huế sẽ xử phạt vi phạm đồng bộ trên phạm vi toàn tỉnh kể từ ngày 01/11/2019. Các hành vi vi phạm về trật tự an toàn giao thông như: xe dừng, đỗ, đón, trả khách không đúng quy định; xe chạy vào đường cấm, vào thời gian cấm; xe không được phép, không có tuyến vào thành phố; xe chạy sai luồng, sai tuyến, xe vượt đèn đỏ, xe quá khổ, xe quá tải; lấn chiếm lòng, lề đường trái phép, thi công công trình trên đường bộ không đảm bảo an toàn giao thông... Các hành vi này được ghi nhận trực tiếp từ hệ thống ghi nhận hình ảnh do Trung tâm Giám sát, điều hành đô thị thông minh quản lý (bao gồm cả các thiết bị ghi nhận hình ảnh của các tổ chức, cá nhân, đơn vị khác đầu tư nhưng đã được phép đấu nối vào hệ thống giám sát điều hành đô thị thông minh và hình ảnh được các cá nhân, tổ chức phản ánh qua kênh tiếp nhận của Trung tâm Giám sát, điều hành đô thị thông minh đã được Trung tâm xác minh thông tin) theo quy định sẽ bị xử lý và xử phạt hành chính theo quy định của pháp luật.

Qua khảo sát có thể thấy rằng, tình trạng chung tại các thành phố lớn đều đã và đang được trang bị các camera giám sát. Tuy nhiên, chưa có các phần mềm xử lý ảnh chuyên dụng để xử lý dữ liệu từ các camera này hỗ trợ cho các lực lượng chức năng quản lý các vi phạm và tra cứu khi cần. Đặc biệt, trong các trường hợp khẩn cấp, với các camera được lắp đặt hiện nay chưa thể truy vết các xe vi phạm khi lưu hành qua các trạm quan sát của camera. Do đó, rất cần thiết xây dựng hệ thống nhận diện biển số xe để giám sát được các xe lưu thông trên địa bàn một cách trực quan và chính xác, hỗ trợ các cơ quan chức năng truy vết được các xe khi cần, nâng cao hiệu quả quản lý.

1.2 Khái niệm về nhận dạng biển số xe

1.2.1 Khái niệm

Hệ thống nhận dạng biển số xe tự động (Automatic License Plate Recognition- ALPR) là hệ thống có khả năng phân tích hình ảnh và xác định biển số xe thông qua video, thiết bị ghi hình và hình ảnh. Sau cùng là xác định các thông tin như: chủ sở hữu xe, theo dõi xe với tốc độ chậm,...



Hình 1.2. Hệ thống nhận dạng biển số xe tự động

Ứng dụng nhận dạng biển số xe là ứng dụng có khả năng phân tích hình ảnh và xác định biển số xe từ các hình ảnh chụp được từ các thiết bị thu hình. Nguồn hình ảnh cho ứng dụng có rất nhiều. Và phát triển, hình ảnh được trực tiếp thu nhận từ camera.

Có nhiều cách thức khác nhau để phân loại các ứng dụng nhận dạng biển số xe. Một trong những cách đơn giản là phân loại ứng dụng nhận dạng biển số xe thông qua mục đích sử dụng. Có thể chia ứng dụng nhận dạng biển số xe thành hai loại sau [1]:

Loại 1: Giới hạn vùng nhìn

- *Đầu vào:* Ảnh thu trực tiếp từ các thiết bị ghi nhận ảnh kỹ thuật số. Ảnh được ghi nhận thường chỉ giới hạn trong vùng có biển số xe.
- *Nguyên lý hoạt động:* Các phương tiện giao thông phải chạy với một tốc độ đủ chậm để máy ghi nhận hình ảnh có thể thu được ảnh vùng biển số xe.
- *Ứng dụng:* Những ứng dụng nhận dạng biển số xe loại này thường được dung tại các trạm kiểm soát, các trạm thu phí, các bãi gửi xe tự động, các trạm gác cổng.

Loại 2: Không giới hạn vùng nhìn:

- *Đầu vào:* Ảnh đầu vào thu được từ các thiết bị ghi hình tự động, không phụ thuộc vào góc độ, các đối tượng xung quanh, ảnh không cần bắt buộc chỉ chụp vùng chứa biển số xe, mà có thể ảnh tổng hợp như chứa thêm các đối tượng như người, cây, đường phố..., miễn là vùng biển số xe phải đủ rõ để có thể thực hiện nhận dạng được các ký tự trong vùng đó.
- *Nguyên lý hoạt động:* Do đặc tính không giới hạn vùng nhìn mà ảnh đầu vào có thể thu được từ một thiết bị ghi hình (camara, máy ảnh...). Và do đó, công việc đầu tiên là dò tìm trong ảnh, để xác định đúng vùng nào là biển số xe. Sau đó, thực hiện tách vùng và nhận dạng. Cuối cùng tùy thuộc vào mục đích sử dụng mà kết quả nhận dạng được truyền đi hay lưu trữ để phục vụ nhu cầu của người dùng cuối.
- *Ứng dụng:* Vì không phụ thuộc vào hình ảnh thu được nên có thể dùng ứng dụng tại nhiều nơi như tại những nơi điều tiết giao thông,

tại các vị trí nhạy cảm của giao thông như ngã ba, ngã tư đường giao nhau. Kiểm soát, nhận diện những hành vi vi phạm an toàn giao thông.

1.2.3. Ứng dụng của hệ thống nhận dạng biển số xe:

Hệ thống nhận dạng biển số xe được xây dựng nhằm mục đích giám sát, kiểm soát các phương tiện. Dưới đây chúng ta đề cập đến một số ứng dụng phổ biến đối với hệ thống nhận dạng biển số xe:

- Thu phí giao thông: Lắp đặt hệ thống “Nhận dạng biển số xe” tại các trạm thu phí nhằm hỗ trợ hoặc tự động hóa công tác thu phí.
- Kiểm soát xe tại các cổng ra vào , bãi đỗ xe
- Giám sát giao thông: Các camera được lắp tại các vị trí nhạy cảm trên đường giám sát các phương tiện tham gia giao thông, nhận diện lỗi sai theo dõi phương tiện.

1.3. Các kỹ thuật nhận dạng biển số xe:

Hệ thống ALPR hoạt động theo ba bước, bước đầu tiên là chụp ảnh giao thông, phương tiện tham gia giao thông, bước thứ hai là nhận diện và trích xuất biển số trong ảnh và bước cuối cùng là sử dụng kỹ thuật phân đoạn hình ảnh (Image segmentation) để nhận từng ký tự và nhận dạng ký tự quang học (OCR) để xác định các ký tự riêng lẻ với sự trợ giúp của cơ sở dữ liệu được lưu trữ cho mỗi ký tự chữ và số [2].

Không có một hệ thống ALPR nào có thể nhận dạng chính xác 100%. Điều đó phụ thuộc vào nhiều yếu tố như thời tiết, độ sáng, góc của camera tới xe,... Một số yếu tố ảnh hưởng đến độ chính xác của hệ thống là:

- Độ phân giải của ảnh kém hoặc ảnh bị mờ.
- Điều kiện ánh sáng yếu, bị phản chiếu hoặc che bóng.
- Các đối tượng có dạng tương tự như biển số xe ở ngoại cảnh.
- Sự khác nhau về cấu trúc biển số xe của mỗi nước

1.3.1. Nhận dạng biển số xe sử dụng kỹ thuật xử lý ảnh

Hai nhiệm vụ cơ bản của quá trình xử lý ảnh là nâng cao chất lượng thông tin hình ảnh và xử lý số liệu cung cấp cho các quá trình khác trong đó có việc ứng dụng thị giác vào điều khiển . Người lập trình sẽ tác động các thuật toán tương ứng lên dữ liệu ảnh nhằm thay đổi cấu trúc ảnh phù hợp nhằm tách đối tượng ra khỏi bức ảnh , cụ thể trong bài toán là biển số xe, sau đó phân vùng ký tự để nhận diện từng ký tự. Đầu ra là vùng biển số xe đã tách và chuỗi ký tự biển số xe đã được xử lý và nhận diện [3].

- **Ưu điểm:** Đơn giản, xử lý nhanh đối với những ảnh chỉ chứa vùng biển số xe với độ chính xác cao

- **Nhược điểm:** Khi ảnh có thêm nhiều đối tượng không phải là vùng biển số xe, chẳng hạn là ảnh chụp tổng quát gồm cả cảnh vật bên ngoài thì cách tiếp cận này trở nên không hiệu quả. Độ phức tạp tính toán khá cao. Khi ảnh có thêm nhiều đối tượng khác thì khối lượng tính toán tăng lên rất nhiều .Vì vậy phương pháp này chỉ hiệu quả đối với hệ thống trạm thu phí, trạm gác cổng, gửi xe tự động.

1.3.2. Nhận dạng biển số xe sử dụng kỹ thuật học máy.

Sử dụng kỹ thuật học máy để đưa ra được các vùng chứa biển số. Từ các vùng biển số xác định được, ta sử dụng các kỹ thuật nhận diện ký tự để xác định chuỗi ký tự trong biển số đó [1].

- **Ưu điểm:** Nhận diện được nhiều biển số có trong một khung hình, độ chính xác cao, xử lý nhanh.

- **Nhược điểm:** Yêu cầu lượng lớn tài nguyên tính toán, cần lượng lớn dữ liệu để huấn luyện mô hình nhận dạng.

Trong phạm vi đồ án, tôi sử dụng một số kỹ thuật học máy áp dụng vào việc nhận diện biển số xe.

Kết luận chương 1

Ở chương 1, học viên đã đưa ra khảo sát về các hệ thống giám sát giao thông đã có ở Việt Nam, các phương pháp nhận diện biển số xe đã được sử dụng từ đó làm rõ được tính cấp thiết của việc xây dựng hệ thống. Việc xây dựng một hệ thống quản lý giám sát giao thông sẽ hỗ trợ rất nhiều cho việc quản lý, giám sát góp phần và giảm nhẹ rủi ro trong giao thông, đảm bảo đáp ứng đúng theo hướng phát triển của nhà nước trong việc ứng dụng công nghệ thông tin trong điều hành quản lý. Có nhiều cách tiếp cận đối với bài toán nhận dạng biển số xe dựa trên công nghệ xử lý hình ảnh, tuy nhiên cùng với sự phát triển của công nghệ và trí tuệ nhân tạo, trong khuôn khổ của đề án tốt nghiệp, học viên sử dụng một số kỹ thuật học máy áp dụng vào việc nhận diện biển số xe nhằm mang lại hiệu quả tốt hơn về hiệu năng và độ chính xác khi nhận dạng. Trong chương tiếp theo, học viên sẽ trình bày một số tìm hiểu về cơ sở lý thuyết để phục vụ cho đề án.

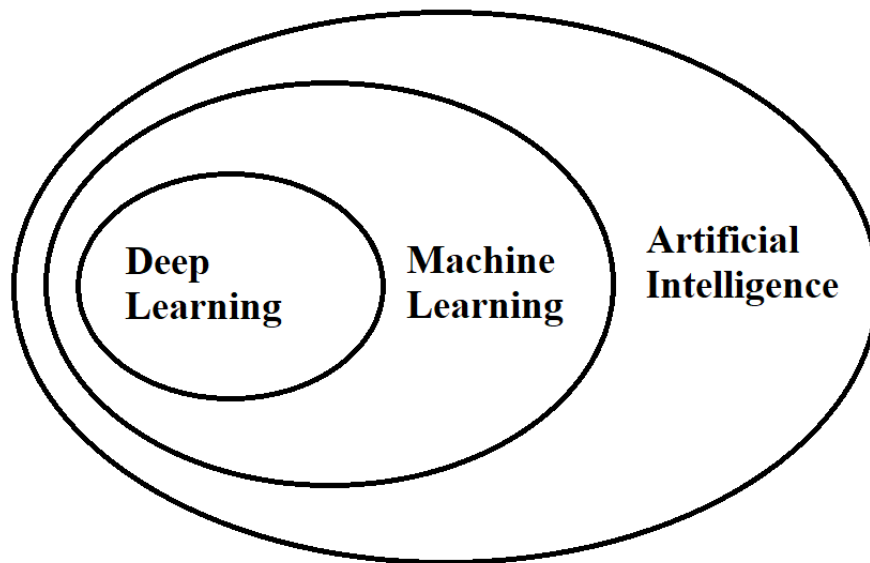
Chương 2

CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về học máy

2.1.1. Tổng quan

AI - Trí tuệ nhân tạo (*Artificial Intelligence*) là các kỹ thuật giúp cho máy tính thực hiện được những công việc của con người chúng ta. Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, ..., chỉ là một vài trong vô vàn những ứng dụng của AI.



Hình 2.1. Mối quan hệ giữa các phương pháp học máy

Trong lĩnh vực AI có một nhánh nghiên cứu về khả năng tự học của máy tính được gọi là ML - Học máy (*Machine Learning*). Hiện nay không có 1 định nghĩa chính thức nào về học máy cả nhưng có thể hiểu rằng nó là các kỹ thuật giúp cho máy tính có thể tự học mà không cần phải cài đặt các luật quyết định. Thường một chương trình máy tính cần các quy tắc, luật lệ để có thể thực thi được một tác vụ nào đó như dán nhãn cho các email là thư rác nếu nội dung email có

chứa từ khoá “quảng cáo”. Nhưng với học máy, các máy tính có thể tự động phân loại các thư rác thành mà không cần chỉ trước bất kỳ quy tắc nào cả.

Một nhánh nhỏ trong học máy gần đây rất được ưu chuộng là DL – Học sâu (*Deep Learning*). Học sâu là kỹ thuật sử dụng các mạng nơ-ron tương tự như các nơ-ron của não người để xây dựng hệ thống học máy. Đây là một sự kết hợp tuyệt vời giữa toán học và khoa học thần kinh. Kết quả của nó mang lại cực kỳ to lớn, có thể coi là khởi nguyên của ngành công nghiệp mới. Tại thời điểm này, hầu hết các tập đoàn lớn cả trong ngành công nghệ lẫn các ngành khác như ô tô, điện tử đều đang tập trung phát triển và ứng dụng kỹ thuật học sâu cho bài toán của mình.

Các giải thuật học máy được phân ra làm các loại chính là:

- Học có giám sát (Supervised Learning)
- Học không giám sát (Unsupervised Learning)
- Học bán giám sát (Semi-supervised learning)
- Học tăng cường (Reinforcement learning)

2.1.2. Học có giám sát (*Supervised Learning*)

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ta có một tập hợp biến đầu vào $X = \{x_1, x_2, \dots, x_N\}$ và một tập hợp nhãn tương ứng $Y = \{y_1, y_2, \dots, y_N\}$, trong đó x_i, y_i là các vector. Các cặp dữ liệu biết trước $(x_i, y_i) \in X \times Y$ được gọi là tập training data (dữ liệu đào tạo). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y :

$$y_i \approx f(x_i), \forall i = 1, 2, \dots, N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu x mới, chúng ta có thể tính được nhãn tương ứng của nó $y=f(x)$.

2.1.3. Học không giám sát (*Unsupervised Learning*)

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào.

2.1.4. Học bán giám sát (*Semi-supervised learning*)

Các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.

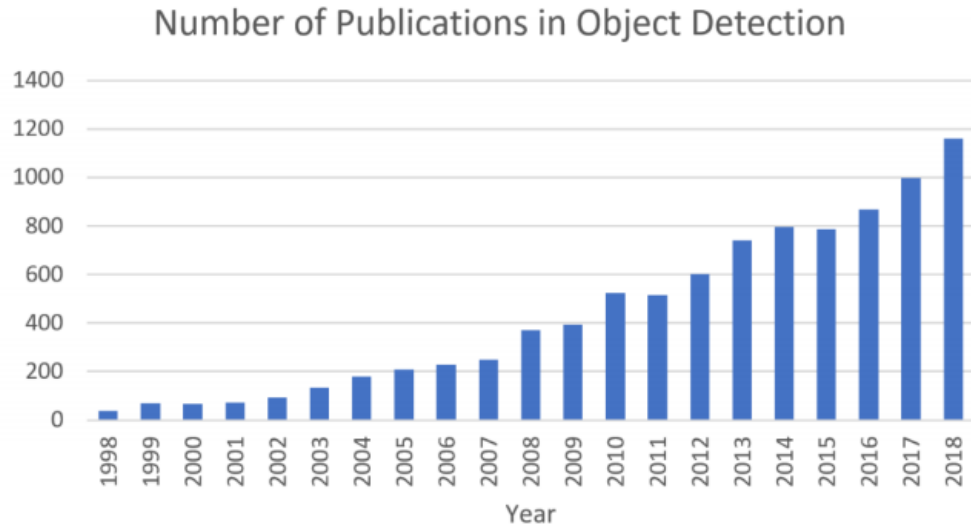
2.2. Bài toán phát hiện đối tượng

Đối với bài nhận diện bằng công nghệ xử lý ảnh nói chung và nhận diện biển số xe nói riêng, một trong những bước đầu tiên, quan trọng nhất là phát hiện được đối tượng. Sau khi phát hiện được đối tượng, sẽ áp dụng các kỹ thuật nâng cao để nhận diện các đặc trưng riêng của đối tượng, từ đó đưa ra đối tượng cần tìm.

Phát hiện đối tượng là một phần quan trọng của thị giác máy tính liên quan đến việc nhận diện các đối tượng trực quan của một lớp nhất định (như con người, động vật hoặc ô tô) trong hình ảnh kỹ thuật số. Mục tiêu của phát hiện đối tượng là phát triển các mô hình và kỹ thuật tính toán cung cấp một trong những thông tin cơ bản nhất cần thiết cho các ứng dụng thị giác máy tính: Đối tượng nào ở đâu? [4]

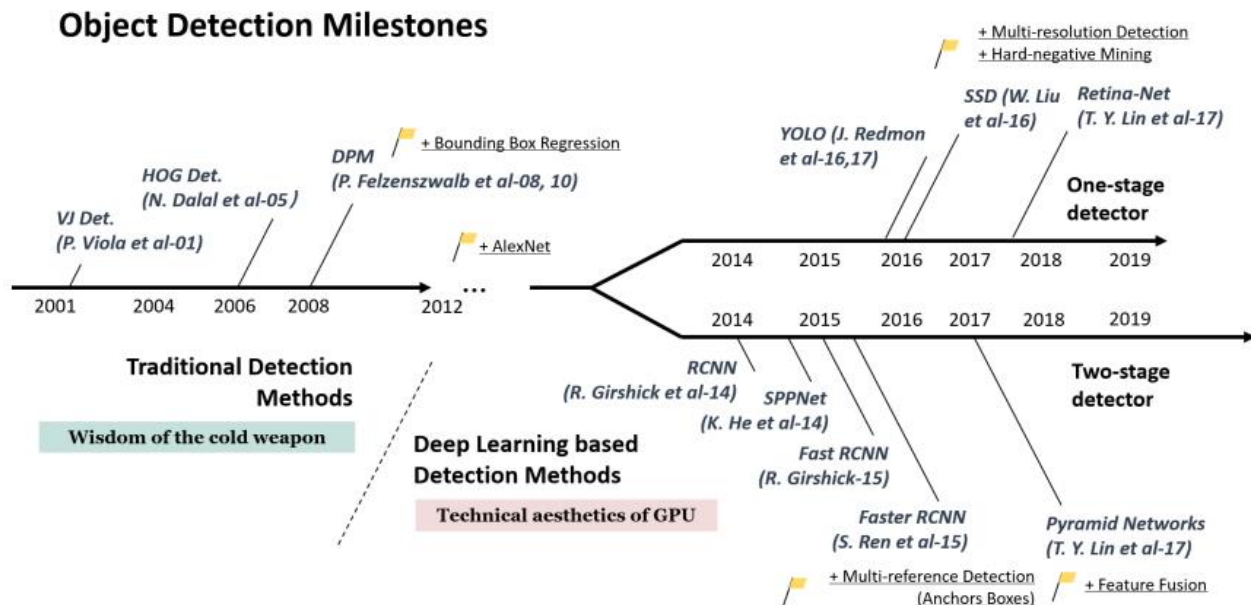
Là một trong những vấn đề cơ bản của thị giác máy tính, phát hiện đối tượng tạo thành nền tảng của nhiều tác vụ thị giác máy tính khác, chẳng hạn như phân đoạn đối tượng (instance segmentation), chú thích hình ảnh (image captioning), theo dõi đối tượng (object tracking),...

Trên quan điểm ứng dụng, phát hiện đối tượng có thể được nhóm thành hai chủ đề nghiên cứu “Phát hiện đối tượng chung” và “các ứng dụng nhận diện”, trong đó mục tiêu trước đây là khám phá các phương pháp nhận diện các loại đối tượng khác nhau trong một khung thống nhất để mô phỏng mắt nhìn và nhận thức của con người và sau này đề cập đến việc nhận diện theo các kịch bản ứng dụng cụ thể, như nhận diện người đi bộ, nhận diện khuôn mặt, nhận diện văn bản, v.v. Trong những năm gần đây, sự phát triển nhanh chóng của các kỹ thuật học sâu đã đưa luồng gió mới vào phát hiện đối tượng, dẫn đến những đột phá đáng chú ý. Phát hiện đối tượng hiện đã được sử dụng rộng rãi trong nhiều ứng dụng trong thế giới thực, chẳng hạn như lái xe tự động, thị giác robot, giám sát video,...



Hình 2.2. Sự gia tăng về số lượng các ấn phẩm liên quan đến phát hiện đối tượng từ 1998 đến 2018

Trong hai thập kỷ qua, mọi người đều thừa nhận rằng phát hiện đối tượng đã trải qua 2 giai đoạn: Thời kỳ phát hiện đối tượng truyền thống (trước năm 2014) và thời kỳ phát hiện đối tượng dựa trên học sâu (sau 2014).



Hình 2.3. Sự phát triển của các thuật toán phát hiện đối tượng

2.2.1. Cách nhận diện truyền thống:

Hầu hết các thuật toán phát hiện đối tượng ban đầu được xây dựng dựa trên các đặc trưng thủ công. Do biểu diễn hình ảnh thiếu hiệu quả vào thời điểm đó,

không có lựa chọn nào khác ngoài việc thiết kế các biểu diễn đặc trưng tinh vi và nhiều kỹ năng tăng tốc để sử dụng với tài nguyên máy tính hạn chế [4].

Các kỹ thuật truyền thống có thể kể đến như : Viola Jones Detector, Histogram of Oriented Gradients (HOG) detector, Deformable Part-based Model (DPM)

2.2.2. Nhận diện 2 giai đoạn dựa trên CNN

Khi hiệu suất của các đặc trưng thủ công trở nên bão hòa, năm 2012, thế giới chứng kiến sự tái sinh của mạng thần kinh tích chập. Mạng tích chập sâu có thể học các biểu diễn đặc trưng mạnh mẽ của một hình ảnh, một câu hỏi tự nhiên là liệu chúng ta có thể đưa nó vào phát hiện đối tượng không? R. Girshick đã dẫn đầu để phá vỡ các bế tắc trong năm 2014 bằng cách đề xuất Regions CNN (RCNN) để phát hiện đối tượng. Kể từ đó, phát hiện đối tượng bắt đầu phát triển với tốc độ chưa từng thấy. Trong thời đại của học sâu, việc phát hiện đối tượng có thể được nhóm thành hai thể loại: Nhận diện hai giai đoạn và nhận diện một giai đoạn, trong đó trước đây là một quá trình xử lý thô đến tinh vi trong khi sau này là hoàn thành trong một bước [4].

2.2.2.1. Region CNN (RCNN)

Ý tưởng đằng sau RCNN rất đơn giản: Bắt đầu bằng việc trích xuất một tập hợp các đề xuất đối tượng (object candidate boxes) bằng cách tìm kiếm chọn lọc. Sau đó, mỗi đề xuất được định cỡ lại thành hình ảnh có kích thước cố định và được đưa vào mô hình CNN được train trên ImageNet (AlexNet) để trích xuất các đặc trưng. Cuối cùng, các phân loại SVM tuyến tính được sử dụng để dự đoán sự hiện diện của một đối tượng trong từng khu vực và để nhận ra các loại đối tượng. RCNN mang lại hiệu suất tăng đáng kể trên VOC07, với sự cải thiện lớn về độ chính xác trung bình (mAP) trung bình từ 33,7% lên 58,5%.

Mặc dù RCNN đã đạt được tiên bộ lớn, nhưng nhược điểm của nó là rõ ràng: sự tính toán đặc trưng dư thừa trên một số lượng lớn các đề xuất chồng chéo (hơn 2000 hộp từ một hình ảnh) dẫn đến tốc độ nhận diện cực kỳ chậm (14

giây cho mỗi hình ảnh với GPU). Cùng năm đó, SPPNet đã được đề xuất và đã khắc phục vấn đề này.

2.2.2.2. Spatial Pyramid Pooling Network (SPPNet)

Năm 2014, Kaiming He đã đề xuất Mạng tổng hợp Kim tự tháp không (SPPNet). Các mô hình CNN trước đây yêu cầu đầu vào có kích thước cố định, ví dụ: hình ảnh 224x224 cho AlexNet. Đóng góp chính của SPPNet là việc giới thiệu lớp Spatial Pyramid Pooling (SPP), cho phép CNN tạo ra một đại diện có độ dài cố định bất kể kích thước của hình ảnh / vùng quan tâm mà không thay đổi kích thước. Khi sử dụng SPPNet để phát hiện đối tượng, các bản đồ đặc trưng chỉ có thể được tính toán từ toàn bộ hình ảnh một lần và sau đó có thể tạo các biểu diễn cố định của các vùng tùy ý để đào tạo các máy dò, giúp tránh tính toán nhiều lần các tính năng tích chập. SPPNet nhanh hơn R-CNN hơn 20 lần mà không giảm độ chính xác (VOC07 mAP = 59,2%).

2.2.2.3. Fast RCNN

Vào năm 2015, R. Girshick đã đề xuất Fast RCNN, đây là một cải tiến hơn nữa của R-CNN và SPPNet. Fast RCNN cho phép chúng ta đồng thời huấn luyện và bộ hồi quy bounding box trong cùng cấu trúc mạng. Trên tập dữ liệu VOC07, Fast RCNN đã tăng mAP từ 58,5% (RCNN) lên 70,0% trong khi với tốc độ nhận diện nhanh hơn 200 lần so với R-CNN.

2.2.2.4. Faster RCNN

Năm 2015, S. Ren và cộng sự. đề xuất Faster RCNN ngay sau Fast RCNN. Faster RCNN là mạng nhận diện đầu cuối và là gần thời gian thực đầu tiên (COCO mAP = 42,7%, COCO mAP = 21,9%, VOC07 mAP = 73,2%, VOC12 mAP = 70,4%, 17 khung hình / giây với ZFNet [45]). Đóng góp chính của Faster-RCNN là việc giới thiệu Mạng đề xuất khu vực (RPN). Từ R-CNN đến Faster RCNN, hầu hết các khối riêng lẻ của một hệ thống phát hiện đối tượng, ví dụ: proposal detection, feature extraction, bounding box regression, ..., đã dần dần được tích hợp vào một learning framework từ đầu đến cuối thống nhất.

Mặc dù Faster RCNN vượt qua nút thắt tốc độ của Fast RCNN, vẫn có sự dư thừa tính toán ở giai đoạn nhận diện tiếp theo. Sau đó, một loạt các cải tiến đã được đề xuất, bao gồm RFCN và Light head RCNN .

2.2.2.5. Feature Pyramid Networks

Năm 2017, T.-Y. Lin và cộng sự. Mạng đặc trưng Kim tự tháp (FPN) được đề xuất trên cơ sở Faster RCNN. Trước FPN, hầu hết các mạng chỉ chạy nhận diện trên lớp trên cùng của mạng. Mặc dù các đặc trưng trong các lớp sâu hơn của CNN có lợi cho nhận dạng loại, nhưng nó không có lợi cho việc phân vùng các đối tượng. Cuối cùng, một kiến trúc topdown với các kết nối bên được phát triển trong FPN để xây dựng ngữ nghĩa cấp cao ở mọi quy mô. Do CNN tự nhiên hình thành một kim tự tháp đặc trưng thông qua sự lan truyền về phía trước của nó, FPN cho thấy những tiến bộ tuyệt vời để nhận diện các vật thể với nhiều kích cỡ khác nhau. Sử dụng FPN trong hệ thống R-CNN cơ bản, đạt được kết quả nhận diện trên bộ dữ liệu của MSCOCO (COCO mAP @ .5 = 59.1%, COCO mAP @ [. 5 ,. 95] = 36,2%). Giờ đây, FPN đã trở thành một khối xây dựng cơ bản của nhiều mạng sau này.

2.2.3 Nhận diện một giai đoạn dựa trên CNN

2.2.3.1. You Look Only Once (YOLO)

YOLO được đề xuất bởi R. Joseph và cộng sự vào năm 2015. Đây là mạng một giai đoạn đầu tiên trong kỷ nguyên học sâu. YOLO cực kỳ nhanh: phiên bản nhanh của YOLO chạy ở tốc độ 155fps với VOC07 mAP = 52,7%, trong khi phiên bản nâng cao của nó chạy ở tốc độ 45fps với VOC07 mAP = 63,4% và VOC12 mAP = 57,9%. YOLO là tên viết tắt của You Look Only Once. Có thể thấy từ tên của nó, các tác giả đã hoàn toàn từ bỏ mô hình nhận diện trước đó về nhận diện đề xuất + xác minh. Thay vào đó, nó tuân theo một triết lý hoàn toàn khác: áp dụng một mạng lưới thần kinh duy nhất cho hình ảnh đầy đủ. Mạng này chia hình ảnh thành các vùng và dự đoán các hộp giới hạn và xác suất cho từng vùng đồng thời. Sau đó, R. Joseph đã thực hiện một loạt các cải tiến trên cơ sở YOLO và đã

đề xuất các phiên bản v2 và v3, cải thiện hơn nữa độ chính xác nhận diện trong khi vẫn giữ tốc độ nhận diện rất cao [4].

Mặc dù có sự cải thiện lớn về tốc độ nhận diện, YOLO bị giảm độ chính xác với các mạng hai giai đoạn, đặc biệt là đối với một số vật thể nhỏ. Các phiên bản tiếp theo của YOLO và SSD được đề xuất sau đã chú ý hơn đến vấn đề này.

2.2.3.2 Single Shot MultiBox Detector (SSD)

SSD được đề xuất bởi W. Liu và cộng sự vào năm 2015. Đây là mạng nhận diện một giai đoạn thứ hai trong kỷ nguyên học sâu. Đóng góp chính của SSD là sự ra đời của các kỹ thuật nhận diện đa tham chiếu và đa độ phân, giúp cải thiện đáng kể độ chính xác nhận diện của mạng nhận diện một giai đoạn, đặc biệt là đối với một số đối tượng nhỏ. SSD có lợi thế về cả tốc độ nhận diện và độ chính xác (VOC07 mAP = 76,8%, VOC12 mAP = 74,9%, COCO mAP @ .5 = 46,5%, mAP @ [. 5, .95] = 26.8%, phiên bản nhanh chạy ở tốc độ 59fps). Sự khác biệt chính giữa SSD và bất kỳ máy dò trước nào là cái trước nhận diện các đối tượng của 5 thang đo khác nhau trên các lớp khác nhau của mạng, trong khi các lớp sau chỉ chạy nhận diện trên các lớp trên cùng của chúng [4].

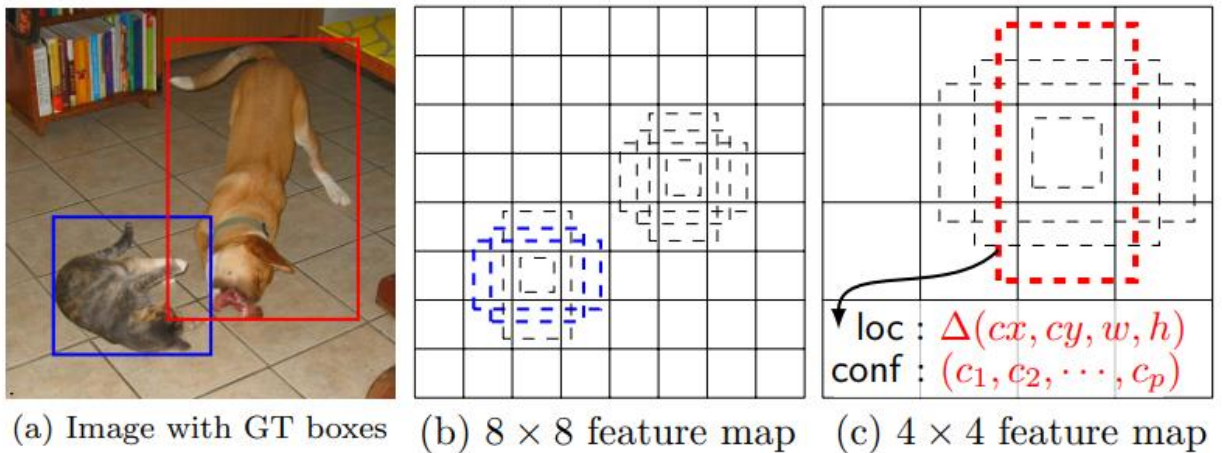
2.2.3.3 RetinaNet

Mặc dù có tốc độ cao và đơn giản, các mạng nhận diện một giai đoạn đã theo sau độ chính xác của mạng nhận diện hai giai đoạn trong nhiều năm. T.-Y. Lin và cộng sự đã nhận diện ra những lý do đằng sau và đề xuất RetinaNet vào năm 2017. Họ tuyên bố rằng sự mất cân bằng lớp nền trước cực đoan gặp phải trong quá trình train mạng dày đặc là nguyên nhân chính. Cuối cùng, một chức năng mất mới có tên là “focal loss” đã được giới thiệu trong RetinaNet bằng cách định hình lại cross entropy loss tiêu chuẩn để máy dò sẽ tập trung hơn vào các ví dụ khó, phân loại sai trong quá trình train. “Focal loss” cho phép các mạng nhận diện một giai đoạn đạt được độ chính xác tương đương của mạng nhận diện hai giai đoạn trong khi vẫn duy trì tốc độ nhận diện rất cao. (COCO mAP @ .5 = 59.1%, mAP @ [. 5, .95] = 39.1%) [4].

2.3. Mô hình SSD Mobilenet

SSD (Single Shot MultiBox Detector) là một thuật toán phổ biến trong phát hiện đối tượng. Nó nhanh hơn Faster RCNN. Trong phần này chúng ta sẽ tìm hiểu kiến trúc mạng SSD, kết hợp với Mobilenet làm mạng cơ sở để phát hiện đối tượng nhanh hơn [5].

2.3.1 SSD (Single Shot MultiBox Detector)

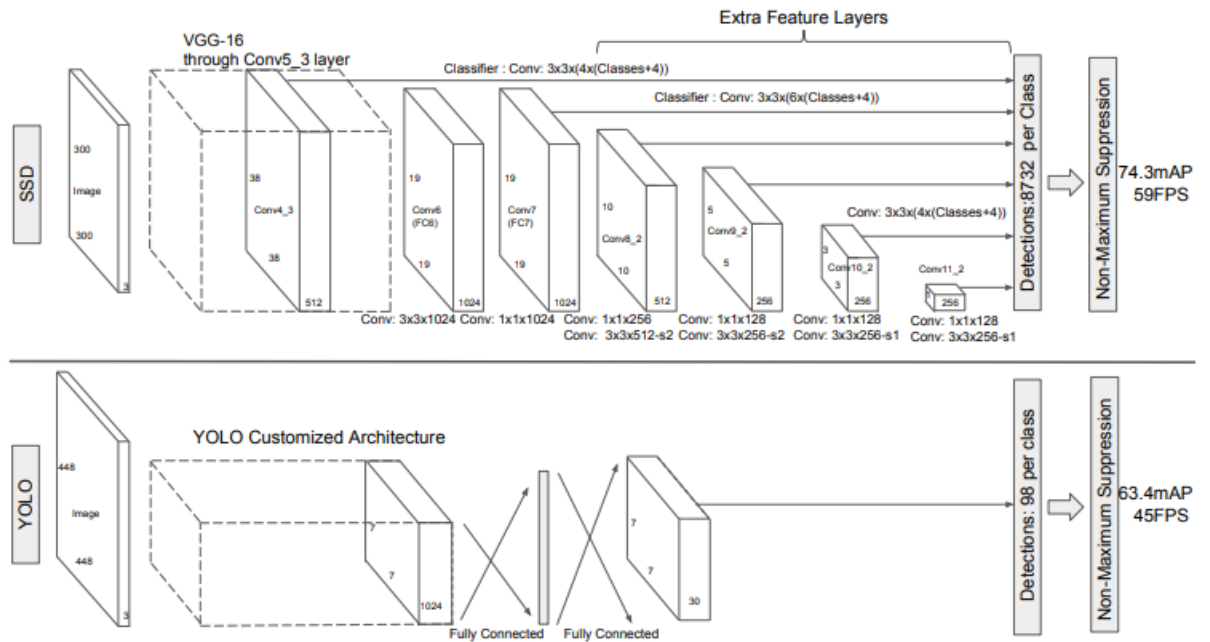


Hình 2.4. Mô hình SSD

a. SSD chỉ cần một ảnh đầu vào và các ground truth boxes cho mỗi đối tượng trong quá trình train. Theo kiểu tích chập, chúng ta đánh giá một tập hợp nhỏ (ví dụ 4) các hộp mặc định có tỷ lệ khung hình khác nhau ở mỗi vị trí trong một số bản đồ đặc trưng với các tỷ lệ khác nhau (ví dụ: 8×8 và 4×4 ở (b) và (c)). Đối với mỗi hộp mặc định, chúng ta dự đoán cả độ lệch hình và độ tin cậy cho tất cả các loại đối tượng ((c_1, c_2, \dots, c_p)). Trong thời gian train, trước tiên chúng ta ghép các hộp mặc định này với các GT boxes. Ví dụ, chúng ta đã ghép hai hộp mặc định với con mèo và một hộp với con chó, được coi là dương và phần còn lại là âm. Loss của mô hình là tổng số có trọng số giữa localization loss (ví dụ: Smooth L1) và confidence loss (ví dụ: Softmax).

Cách tiếp cận của SSD dựa trên mạng tích chập chuyển tiếp nguồn cấp dữ liệu tạo ra một bộ sưu tập các bounding box kích thước cố định và điểm số cho sự hiện diện của của lớp đối tượng trong các hộp đó, sau đó là bước triệt tiêu không tối đa để tạo ra các nhận diện cuối cùng. Các lớp mạng ban đầu dựa trên kiến trúc tiêu chuẩn được sử dụng để phân loại hình ảnh chất lượng cao (được cắt bớt trước bởi các lớp phân loại), mà chúng ta sẽ gọi là mạng cơ sở. Sau đó, chúng ta thêm cấu trúc phụ trợ vào mạng để tạo ra các nhận diện với các tính năng chính:

- Bản đồ đặc trưng đa tỷ lệ để nhận diện: Chúng ta thêm các lớp đặc trưng tích chập vào cuối mạng cơ sở. Các lớp này giảm kích thước dần dần và cho phép dự đoán nhận diện ở nhiều tỷ lệ. Mô hình tích chập để dự đoán nhận diện là khác nhau đối với mỗi lớp đặc trưng (Overfeat và YOLO hoạt động trên bản đồ đặc trưng tỷ lệ đơn).
- Công cụ dự đoán tích chập để nhận diện: Mỗi lớp đặc trưng được thêm vào (hoặc tùy chọn một lớp đặc trưng hiện có từ mạng cơ sở) có thể tạo ra một bộ dự đoán nhận diện cố định bằng cách sử dụng một bộ lọc tích chập. Chúng được biểu thị trên đỉnh của kiến trúc mạng SSD trong Hình 2. Đối với lớp đặc trưng có kích thước $m \times n$ với p kênh, yếu tố cơ bản để dự đoán các tham số của nhận diện tiềm năng là kernel nhỏ $3 \times 3 \times p$ tạo ra một điểm cho một loại hoặc hình dạng bù tương ứng với tọa độ hộp mặc định. Tại mỗi vị trí $m \times n$ nơi kernel được áp dụng, nó tạo ra một giá trị đầu ra. Các giá trị đầu ra bù của bounding box được đo tương ứng với vị trí box mặc định so với từng vị trí bản đồ đặc trưng (kiến trúc của YOLO sử dụng lớp fully connected thay vì bộ lọc tích chập cho bước này).



Hình 2.5. So sánh giữa hai mô hình SSD và YOLO

Mô hình SSD bổ sung một số lớp tính năng vào cuối mạng cơ sở, dự đoán độ lệch cho các hộp mặc định có tỷ lệ và tỷ lệ khung hình khác nhau và các tâm sự liên quan của chúng.

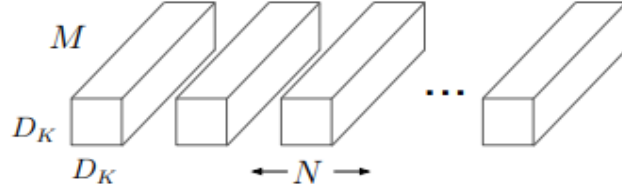
SSD có kích thước đầu vào 300×300 vượt trội đáng kể so với YOLO 448×448 về độ chính xác trong thử nghiệm VOC2007 đồng thời cải thiện tốc độ.

- Các hộp mặc định và tỷ lệ khung hình: Chúng ta liên kết một tập hợp các bounding box mặc định với mỗi ô bản đồ đặc trưng, cho nhiều bản đồ đặc trưng ở đầu mạng. Các hộp mặc định xếp bản đồ đặc trưng theo cách tích chập, sao cho vị trí của mỗi hộp tương ứng với ô tương ứng của nó được cố định. Ở mỗi ô bản đồ đặc trưng, chúng ta dự đoán các độ lệch tương ứng với các hình dạng hộp mặc định trong ô, cũng như điểm số trên mỗi lớp cho biết sự hiện diện của một lớp thực thể trong mỗi ô đó. Cụ thể, với mỗi k hộp tại một vị trí nhất định, chúng ta tính điểm của lớp c và 4 độ lệch so với hộp mặc định ban đầu. Điều này dẫn đến tổng số $(c + 4)$ k bộ lọc được áp dụng xung quanh từng vị trí trong bản đồ đặc trưng, mang lại $(c + 4) kmn$ đầu ra cho bản đồ đặc trưng $m \times n$. Các hộp mặc định tương tự như các anchor boxes được sử dụng trong Faster R-CNN, tuy nhiên nó

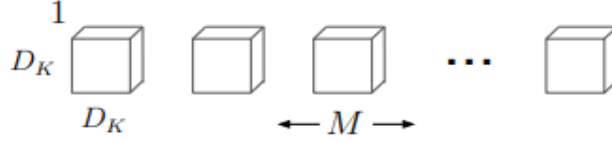
áp dụng cho một số bản đồ đặc trưng có độ phân giải khác nhau. Cho phép các hình dạng hộp mặc định khác nhau trong một số bản đồ đặc trưng cho phép nó phân tách hiệu quả không gian của các hình dạng hộp đầu ra.

2.3.2. *MobileNet*

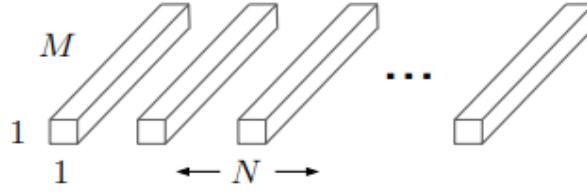
Mô hình MobileNet dựa trên các cấu trúc có thể phân tách theo chiều sâu, là một dạng của các cấu trúc được nhân tố hóa, tạo thành một tích chập tiêu chuẩn thành một tích chập theo chiều sâu và một tích chập 1×1 được gọi là tích chập theo chiều. Đối với MobileNet, tích chập theo chiều sâu áp dụng một bộ lọc duy nhất cho mỗi kênh đầu vào. Tích chập theo chiều sau đó áp dụng tích chập 1×1 để kết hợp các đầu ra tích chập theo chiều sâu. Một tích chập tiêu chuẩn cả hai bộ lọc và kết hợp các đầu vào thành một bộ đầu ra mới trong một bước. Sự tích chập có thể phân tách theo chiều sâu chia tách thành hai lớp, một lớp riêng để lọc và một lớp riêng để kết hợp. Yếu tố này có tác dụng làm giảm đáng kể tính toán và kích thước mô hình. Hình 2 cho thấy cách tích chập tiêu chuẩn 2(a) được xác định thành tích chập 2(b) theo chiều sâu và tích chập tích chập theo chiều sâu 1×1 2(c) [6].



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



Hình 2.6. Các bộ lọc tích chập tiêu chuẩn

Trong (a) được thay thế bằng hai lớp: tích chập theo chiều sâu trong (b) và tích chập theo chiều trong (c) để xây dựng bộ lọc tách sâu.

Một lớp chập tiêu chuẩn lấy đầu vào là bản đồ đặc trưng \mathbf{F} : $D_F \times D_F \times M$ và tạo ra bản đồ đặc trưng \mathbf{G} : $D_F \times D_F \times N$ trong đó D_F là chiều rộng và chiều cao không gian của bản đồ đặc trưng, M là số kênh đầu vào (độ sâu đầu vào), D_G là chiều rộng và chiều cao không gian của bản đồ đặc trưng đầu ra vuông và N là số kênh đầu ra (độ sâu đầu ra).

Lớp tích chập tiêu chuẩn được tham số hóa bởi nhân tích chập \mathbf{K} có kích thước $D_K \times D_K \times M \times N$ trong đó D_K là kích thước không gian của nhân được giả định là hình vuông và M là số kênh đầu vào và N là số kênh đầu ra như được xác định trước đó . Bản đồ đặc trưng đầu ra cho tích chập tiêu chuẩn giả sử bước một và phân padding được tính là:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+j-1,l+j-1,m} \quad (1)$$

Kết quả tiêu chuẩn chi phí tính toán:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2)$$

Trong đó chi phí tính toán phụ thuộc nhiều vào số lượng kênh đầu vào M , số lượng kênh đầu ra N kích thước nhân $D_k \times D_k$ và kích thước bản đồ đặc trưng $D_F \times D_F$. Các mô hình MobileNet giải quyết từng vấn đề này và các tương tác của chúng. Đầu tiên, nó sử dụng các cấu trúc phân tách theo chiều sâu để phá vỡ sự tương tác giữa số lượng kênh đầu ra và kích thước của nhân.

Tích chập tiêu chuẩn có tác dụng lọc các đặc trưng dựa trên các nhân tích chập và các tính năng kết hợp để tạo ra một đại diện mới. Các bước lọc và kết hợp có thể được chia thành hai bước thông qua việc sử dụng các cấu trúc có hệ số được gọi là các cấu trúc phân tách theo chiều sâu để giảm đáng kể chi phí tính toán.

Cấu trúc MobileNet được xây dựng trên các cấu trúc có thể phân tách theo chiều sâu như đã đề cập trong phần trước ngoại trừ lớp đầu tiên là một tổ hợp đầy đủ. Bằng cách xác định mạng theo các thuật ngữ đơn giản như vậy, chúng ta có thể dễ dàng khám phá các cấu trúc liên kết mạng để tìm một mạng tốt. Kiến trúc MobileNet được định nghĩa trong Bảng 1. Tất cả các lớp được theo sau bởi một batchnorm và ReLU phi tuyến ngoại trừ lớp được kết nối đầy đủ cuối cùng không có phi tuyến và đưa vào lớp softmax để phân loại. Hình 2.6 tương phản một lớp với độ chụm thông thường, độ chụm và độ phi tuyến ReLU với lớp nhân tổ với độ chập theo chiều sâu, độ chụm theo chiều dọc 1×1 cũng như độ chụm và ReLU sau mỗi lớp chập. Lấy mẫu xuống được xử lý với tích chập trong các cấu trúc theo chiều sâu cũng như trong lớp đầu tiên. Việc gộp chung trung bình cuối cùng làm giảm độ phân giải không gian xuống 1 trước khi lớp được kết nối đầy đủ. Đếm theo chiều sâu và theo tích chập như các lớp tách biệt, MobileNet có 28 lớp.

Không đủ để đơn giản định nghĩa các mạng theo một số lượng nhỏ Mult-Adds. Nó cũng quan trọng để đảm bảo các hoạt động này có thể được thực hiện một cách hiệu quả. Ví dụ, các hoạt động ma trận thưa thớt không có cấu trúc

thường không nhanh hơn các hoạt động ma trận dày đặc cho đến khi mức độ thưa thớt rất cao. Cấu trúc mô hình của chúng ta đặt gần như tất cả các tính toán thành các kết cấu 1×1 dày đặc. Điều này có thể được thực hiện với các hàm nhân ma trận tổng quát (GEMM) được tối ưu hóa cao. Thông thường các tích chập được thực hiện bởi GEMM nhưng yêu cầu sắp xếp lại ban đầu trong bộ nhớ gọi là im2col để ánh xạ nó tới GEMM. Ví dụ, phương pháp này được sử dụng trong Caffe phổ biến. Các kết quả 1×1 không yêu cầu sắp xếp lại trong bộ nhớ và có thể được thực hiện trực tiếp với GEMM, một trong những thuật toán đại số tuyến tính được tối ưu hóa nhất. MobileNet dành 95% thời gian tính toán của nó trong các kết cấu 1×1 , cũng có 75% các tham số có thể thấy trong Bảng 2.2. Gần như tất cả các tham số bổ sung đều nằm trong lớp được kết nối đầy đủ.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

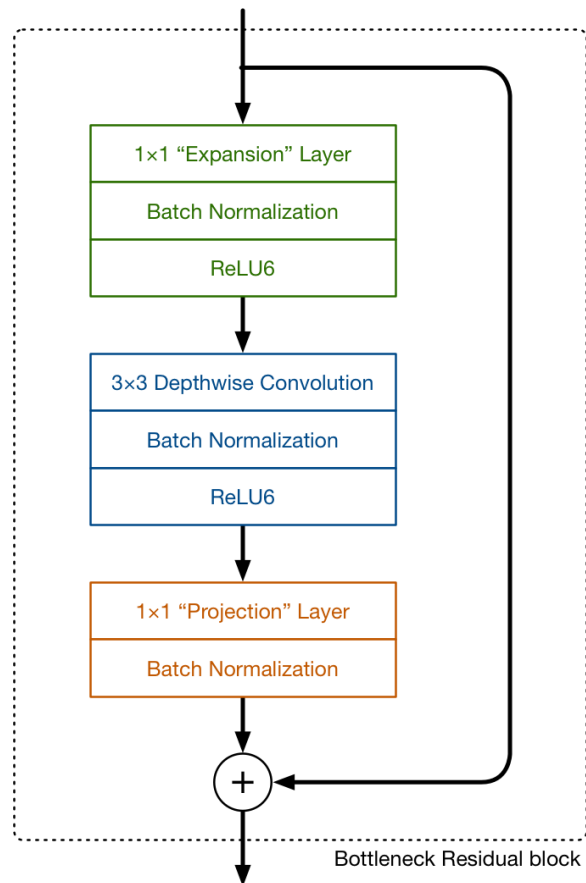
Bảng 2.1. Kiến trúc MobileNet

Các mô hình MobileNet đã được train bằng TensorFlow bằng cách sử dụng RMSprop với gradient descent không đồng bộ tương tự như Inception V3. Tuy nhiên, trái với việc đào tạo các mô hình lớn, chúng ta sử dụng các kỹ thuật tăng cường dữ liệu và chính quy hóa ít hơn bởi vì các mô hình nhỏ ít gặp rắc rối hơn với overfitting. Khi train MobileNets, chúng ta không sử dụng làm mịn nhân và giảm số lượng hình ảnh biến dạng bằng cách giới hạn kích thước của các crop nhỏ được sử dụng trong train Inception lớn. Ngoài ra, chúng ta thấy rằng điều quan trọng là phải đặt rất ít hoặc không phân rã weight (chuẩn hóa l2) trên các bộ lọc theo chiều sâu vì chúng có rất ít tham số trong đó.

Type	Multi-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

Bảng 2.2. Tài nguyên cho mỗi loại tầng

MobileNet V2 vẫn sử dụng các kết cấu có thể phân tách theo chiều sâu, nhưng khối chính của sẽ như sau:



Hình 2.7. Main building block MobileNet V2

Lần này có ba lớp chập trong khối. Hai cái cuối cùng là những cái mà chúng ta đã biết: một tích chập theo chiều sâu, lọc các đầu vào, theo sau là một lớp chập theo chiều 1×1 . Tuy nhiên, lớp 1×1 này hiện có một công việc khác.

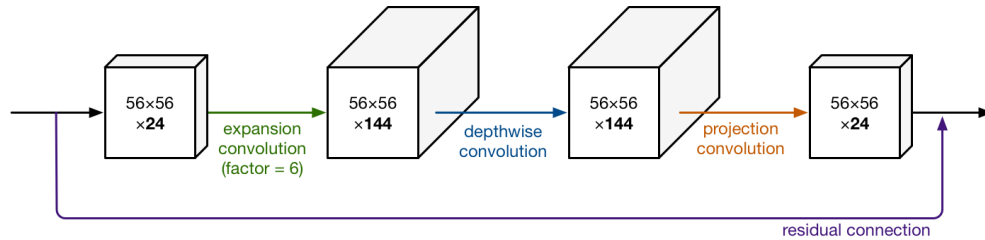
Trong V1, tích chập theo chiều thuận hoặc giữ số lượng kênh giống nhau hoặc nhân đôi chúng. Trong V2 thì ngược lại: nó làm cho số lượng kênh nhỏ hơn. Đây là lý do tại sao lớp này hiện được gọi là lớp chiều - nó chiều dữ liệu với số chiều (kênh) cao thành một tensor với số lượng chiều thấp hơn nhiều.

Ví dụ, lớp chiều sâu có thể hoạt động trên một tensor với 144 kênh, lớp chiều sau đó sẽ co lại chỉ còn 24 kênh. Loại lớp này còn được gọi là lớp nút cổ chai vì nó làm giảm lượng dữ liệu qua mạng.

Lớp đầu tiên là mới trong khối. Đây cũng là một tổ hợp 1×1 . Mục đích của nó là mở rộng số lượng kênh trong dữ liệu trước khi đi vào tích chập sâu. Do

đó, lớp mở rộng này luôn có nhiều kênh đầu ra hơn các kênh đầu vào - nó hoàn toàn ngược lại với lớp chiếu.

Chính xác là bao nhiêu dữ liệu được mở rộng được đưa ra bởi hệ số mở rộng. Đây là một trong những siêu đường kính để thử nghiệm với sự đánh đổi kiến trúc khác nhau. Hệ số mở rộng mặc định là 6. Ví dụ: nếu có một tensor với 24 kênh đi vào một khối, lớp mở rộng trước tiên sẽ chuyển đổi nó thành một tensor mới với $24 \times 6 = 144$ kênh. Tiếp theo, tích chập theo chiều sâu áp dụng các bộ lọc của nó cho tensor 144 kênh đó. Và cuối cùng, lớp chiếu chiếu 144 kênh đã lọc trở lại một số nhỏ hơn, 24.



Hình 2.8. Đầu vào và đầu ra của khối là các tensor chiều thấp

Vì vậy, đầu vào và đầu ra của khối là các tensor chiều thấp, trong khi bước lọc xảy ra bên trong khối được thực hiện trên một tensor chiều cao.

Điều mới thứ hai trong khối xây dựng MobileNet V2, là kết nối còn lại. Điều này hoạt động giống như trong ResNet và tồn tại để trợ giúp dòng chảy của gradient thông qua mạng. (Kết nối còn lại chỉ được sử dụng khi số lượng kênh đi vào khối giống với số lượng kênh ra khỏi nó, điều này không phải lúc nào cũng như vậy vì cứ sau vài khối thì các kênh đầu ra lại tăng lên.)

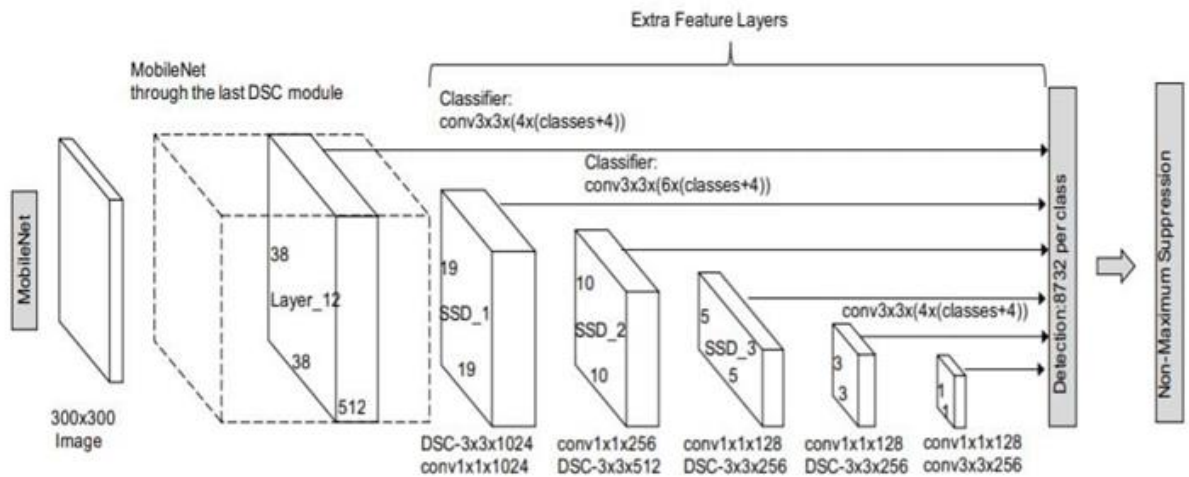
Như thường lệ, mỗi lớp có chuẩn hóa hàng loạt và hàm kích hoạt là ReLU6. Tuy nhiên, đầu ra của lớp chiếu không có hàm kích hoạt được áp dụng cho nó. Vì lớp này tạo ra dữ liệu chiều thấp.

Kiến trúc MobileNet V2 đầy đủ, sau đó, bao gồm 17 khối xây dựng liên tiếp. Tiếp theo là tích chập 1×1 thông thường, lớp pooling trung bình toàn cục

và lớp phân loại. (Chi tiết nhỏ: khối đầu tiên hơi khác một chút, nó sử dụng tích chập 3×3 thông thường với 32 kênh thay vì lớp mở rộng.)

2.3.3. SSD MobileNet

SSD được thiết kế độc lập với mạng cơ sở và do đó, nó có thể chạy trên hầu hết mọi mạng cơ sở, kể cả MobileNet. Thậm chí tốt hơn, MobileNet + SSD sử dụng một biến thể có tên SSDLite sử dụng các lớp có thể phân tách theo chiều sâu thay vì các cấu trúc thông thường cho phần phát hiện đối tượng của mạng. Với SSDLite trên MobileNet, ta có thể dễ dàng nhận được kết quả thực sự theo thời gian thực (ví dụ: 30 FPS trở lên) [7].



Hình

Hình 2.9. Mô hình kiến trúc SSD MobileNet

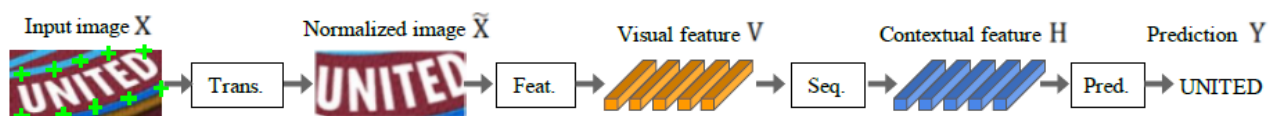
Mô hình SSD MobileNet được sử dụng trong đồ án để nhận diện biển số xe từ hình ảnh trích xuất từ camera.

2.4. Nhận dạng văn bản (Scene Text Recognition)

Nhận dạng văn bản (STR) bao gồm bốn giai đoạn. Do sự tương đồng của STR với các nhiệm vụ thị giác máy tính (ví dụ: phát hiện đối tượng) và các nhiệm vụ dự đoán chuỗi, STR đã được hưởng lợi từ các mạng thần kinh tích chập hiệu

suất cao (CNNs) và mạng thần kinh hồi quy (RNNs). Ứng dụng kết hợp đầu tiên của CNN và RNN cho STR, Mạng thần kinh tái tạo liên tục (CRNN), trích xuất các đặc trưng CNN từ hình ảnh văn bản đầu vào và cấu hình lại chúng bằng RNN để dự đoán chuỗi. Sau CRNN, nhiều biến thể đã được đề xuất để cải thiện hiệu suất. Để chỉnh sửa văn bản tùy ý, ví dụ, các mô đun chuyển đổi đã được đề xuất để chuẩn hóa hình ảnh văn bản. Để xử lý các hình ảnh văn bản phức tạp với các yếu tố tiềm ẩn và yếu tố tiềm ẩn cao (ví dụ: kiểu phong chữ và nền lộn xộn), các trình trích xuất tính năng CNN được cải tiến đã được kết hợp. Ngoài ra, khi mọi người trở nên quan tâm hơn với thời gian suy luận, một số phương pháp thậm chí đã bỏ qua giai đoạn RNN. Để cải thiện dự đoán trình tự ký tự, bộ giải mã dựa trên sự chú ý đã được đề xuất [8]. Bốn giai đoạn bắt nguồn từ các mô hình STR hiện có như sau:

- Transformation (Trans.) Bình thường hóa hình ảnh văn bản đầu vào bằng cách sử dụng Mạng biến đổi không gian (STN) để giảm bớt các giai đoạn xuôi dòng.
- Feature extraction (Feat.) Ánh xạ hình ảnh đầu vào thành một đại diện tập trung vào các thuộc tính có liên quan để nhận dạng ký tự, đồng thời loại bỏ các đặc trưng không liên quan như phong chữ, màu sắc, kích thước và nền.
- Sequence modeling (Seq.) Nắm bắt thông tin theo ngữ cảnh trong một chuỗi các ký tự cho giai đoạn tiếp theo để dự đoán từng ký tự chính xác hơn, thay vì thực hiện độc lập.
- Prediction (Dự đoán) ước tính chuỗi ký tự đầu ra từ các đặc trưng được xác định của hình ảnh.



Hình 2.10. Quá trình nhận diện text từ ảnh chụp

2.4.1. Giai đoạn chuyển đổi

Mô-đun của giai đoạn này biến đổi hình ảnh đầu vào X thành hình ảnh chuẩn hóa \tilde{X} . Hình ảnh văn bản trong các cảnh tự nhiên có nhiều hình dạng khác nhau, như được hiển thị bằng các văn bản cong và nghiêng. Nếu các hình ảnh đầu vào như vậy được cung cấp không thay đổi, giai đoạn trích xuất tính năng tiếp theo cần phải tìm hiểu một biểu diễn bất biến đối với hình học đó. Để giảm bớt gánh nặng này, phép biến đổi thin-plate spline (TPS), một biến thể của mạng biến đổi không gian (STN), đã được áp dụng với tính linh hoạt của nó đối với các tỷ lệ khung hình khác nhau của các dòng văn bản. TPS sử dụng phép nội suy spline trơn tru giữa một tập hợp các *fiducial point*. Chính xác hơn, TPS tìm thấy nhiều điểm *fiducial point* (dấu hiệu ++ màu xanh lục trong Hình 3) tại các điểm bao bọc trên và dưới và bình thường hóa vùng ký tự thành một hình chữ nhật được xác định trước.

2.4.2. Giai đoạn trích xuất đặc trưng

Trong giai đoạn này, CNN trùu tượng một hình ảnh đầu vào (tức là, X hoặc \tilde{X}) và xuất ra bản đồ đặc trưng trực quan $V = \{v_i\}$, $i = 1, \dots, I$ (I là số cột trong bản đồ đặc trưng). Mỗi cột trong bản đồ đặc trưng kết quả của một trình trích xuất tính năng có trường tiếp nhận có thể phân biệt tương ứng dọc theo đường ngang của hình ảnh đầu vào. Các tính năng này được sử dụng để ước tính ký tự trên từng trường tiếp nhận. Ba kiến trúc của VGG, RCNN và ResNet, trước đây được sử dụng làm công cụ trích xuất tính năng cho STR. VGG ở dạng ban đầu bao gồm nhiều lớp chập theo sau là một vài lớp được kết nối đầy đủ. RCNN là một biến thể của CNN có thể được áp dụng đệ quy để điều chỉnh các trường tiếp nhận của nó tùy thuộc vào hình dạng ký tự. ResNet là một CNN với các kết nối còn lại giúp giảm bớt việc đào tạo các CNN tương đối sâu hơn.

2.4.3. Giai đoạn mô hình hóa trình tự

Các đặc trưng được trích xuất từ giai đoạn trích xuất đặc trưng được định hình lại là một chuỗi các đặc trưng V . Nghĩa là, mỗi cột trong bản đồ đặc trưng

$v_i \in V$ được sử dụng làm khung của chuỗi. Tuy nhiên, trình tự này có thể bị thiếu thông tin theo ngữ cảnh. Do đó, sử dụng LSTM hai chiều (BiLSTM) để tạo chuỗi tốt hơn $H = Seq.(V)$ sau giai đoạn trích xuất đặc trưng.

2.4.4. Giai đoạn dự đoán

Trong giai đoạn này, từ đầu vào H , một mô-đun dự đoán một chuỗi các ký tự, (tức là, $Y = y_1, y_2, \dots$). Bằng cách tóm tắt các công việc trước đó, có hai tùy chọn để dự đoán: (1) Phân loại thời gian kết nối (CTC) và (2) dự đoán trình tự dựa trên sự chú ý (Attn). CTC cho phép dự đoán số lượng chuỗi không cố định mặc dù số lượng tính năng cố định được đưa ra. Các phương pháp chính cho CTC là dự đoán một ký tự ở mỗi cột ($h_i \in H$) và sửa đổi chuỗi ký tự đầy đủ thành một dòng ký tự không cố định bằng cách xóa các ký tự và khoảng trống lặp lại. Mặt khác, Attn tự động nắm bắt luồng thông tin trong chuỗi đầu vào để dự đoán chuỗi đầu ra. Nó cho phép một mô hình STR học một mô hình ngôn ngữ cấp ký tự đại diện cho các phụ thuộc của lớp đầu ra.

Mô hình nhận diện này được sử dụng trong đề án để nhận diện chuỗi trong ảnh biển số xe được trích xuất từ ảnh camera qua mô hình SSD MobileNet

2.5. Bản đồ số

2.5.1. Khái niệm về bản đồ số

Bản đồ số là một tập hợp có tổ chức các dữ liệu bản đồ trên thiết bị có khả năng đọc bằng máy tính và được thể hiện dưới dạng hình ảnh bản đồ.

Bản đồ số bao gồm các thành phần cơ bản sau:

- Thiết bị ghi dữ liệu.
- Máy tính,
- Cơ sở dữ liệu,
- Thiết bị thể hiện bản đồ.

Bản đồ số được tổ chức và lưu trữ gọn nhẹ, khác với bản đồ truyền thống ở chỗ: Bản đồ số chỉ là các file dữ liệu ghi trong bộ nhớ máy tính và có thể thể hiện ở dạng hình ảnh giống như bản đồ truyền thống trên màn hình máy tính. Nếu sử dụng các máy vẽ thì ta có thể in được bản đồ trên giấy giống như bản đồ thông thường.

Nhờ các máy tính có khả năng lưu trữ khối lượng thông tin lớn, khả năng tổng hợp, cập nhật, phân tích thông tin và xử lý dữ liệu bản đồ phong phú nên bản đồ số được ứng dụng rộng rãi và đa dạng hơn rất nhiều so với bản đồ giấy.

2.5.2. *Google Map API*

Google Maps Api là công kết nối các tính năng hiện có của Google Map dành cho các lập trình viên để phát triển các ứng dụng bên thứ ba. Google Map Api được google phát triển và tập trung vào 3 mảng chính sau:

- Triển khai Google Map trên các ứng dụng web mà không phải quan tâm công nghệ phát triển đằng sau nó là gì: php, asp.net, java ...
- Tích hợp vào các ứng dụng mobile, cung cấp các tiện ích về chỉ đường, đánh dấu, tính toán khoảng cách, thời gian hoàn thành chuyến đi. Uber và Grab là hai ứng dụng mobile nổi bật cho ví dụ này.
- Các ứng dụng webservice: là tham chiếu để tạo ra các bộ thư viện, dịch vụ cho các ứng dụng khác cần tích hợp

2.5.3. *Direction API*

Directions API là dịch vụ tính toán chỉ đường giữa các vị trí. Có thể tìm kiếm chỉ đường cho một số phương thức di chuyển, bao gồm cả phương tiện, lái xe, đi bộ hoặc đi xe đạp.

Với Directions API, có thể:

- Tìm kiếm chỉ đường cho một số phương thức vận tải, bao gồm vận chuyển, lái xe, đi bộ hoặc đi xe đạp.

- Trả lại chỉ đường nhiều với nhiều kết quả khác nhau bằng cách sử dụng một loạt các điểm tham chiếu.
- Chỉ định nguồn gốc, điểm đến và điểm tham chiếu dưới dạng chuỗi văn bản (ví dụ: "Tây Tiến, Tiền Hải, Thái Bình" hoặc "Tiền Hải, Thái Bình") hoặc dưới dạng vĩ độ / kinh độ hoặc dưới dạng ID địa điểm.

API trả về các tuyến đường hiệu quả nhất khi tính toán chỉ đường. Thời gian đi lại là yếu tố chính được tối ưu hóa, nhưng API cũng có thể tính đến các yếu tố khác như khoảng cách, số lượt rẽ và nhiều yếu tố khác khi quyết định tuyến nào hiệu quả nhất.

a. Thông số bắt buộc

*Origin: Địa chỉ, giá trị vĩ độ / kinh độ văn bản hoặc ID địa điểm mà bạn muốn tính chỉ đường.

- Nếu dùng địa chỉ, Directions API sẽ định vị chuỗi và chuyển đổi nó thành tọa độ vĩ độ / kinh độ để tính chỉ đường. Tọa độ này có thể khác với tọa độ được API mã hóa địa lý trả về, ví dụ như lối vào tòa nhà thay vì trung tâm của nó.

Ví dụ: origin = 234 + Hoàng+ Quốc + Việt + On

- Nếu dùng tọa độ, chúng được sử dụng không thay đổi để tính chỉ đường. Đảm bảo rằng không có khoảng trống nào tồn tại giữa các giá trị vĩ độ và kinh độ.

Ví dụ: origin = 41.43206, -81.38992

-Nếu dùng Place IDs thì phải được bắt đầu bằng place_id: ID địa điểm chỉ có thể được chỉ định nếu yêu cầu bao gồm khóa API hoặc ID ứng dụng API cao cấp của API Google Maps. Có thể truy xuất ID địa điểm từ API mã hóa địa lý và SDK địa điểm (bao gồm cả Tự động điền địa điểm).

Ví dụ: origin = place_id: ChIJ3S-JXmauEmsRUcIaWtf4MzE

*Destination: Địa chỉ, giá trị vĩ độ / kinh độ văn bản hoặc ID địa điểm mà bạn muốn tính chỉ đường. Các tùy chọn cho tham số đích giống với tham số gốc, được mô tả ở trên.

*Key: Khóa API của ứng dụng của bạn. Khóa này xác định ứng dụng của bạn cho mục đích quản lý hạn ngạch. Tìm hiểu cách lấy chìa khóa.

b. Các tham số tùy chọn

*Mode (mặc định để lái xe): Chỉ định phương thức vận tải để sử dụng khi tính chỉ đường.

*Waypoints: Chỉ định một mảng các điểm tham chiếu. Điểm tham chiếu thay đổi tuyến đường bằng cách định tuyến nó qua các vị trí được chỉ định. Điểm tham chiếu được chỉ định dưới dạng tọa độ vĩ độ / kinh độ, một polyline được mã hóa, một ID địa điểm hoặc một địa chỉ sẽ được mã hóa địa lý.

* Alternatives: Nếu được đặt thành true, hãy chỉ định rằng dịch vụ Chỉ đường có thể cung cấp nhiều hơn một thay thế tuyến đường trong phản hồi. Lưu ý rằng việc cung cấp các lựa chọn thay thế tuyến đường có thể làm tăng thời gian phản hồi từ máy chủ. Tính năng này chỉ khả dụng cho các yêu cầu không có điểm tham chiếu trung gian.

*Avoid: Chỉ ra rằng các tuyến đường được tính toán nên tránh các tính năng được chỉ định. Tham số này hỗ trợ các đối số sau:

- Tools: phí cầu đường cho thấy tuyến đường được tính toán nên tránh đường hoặc cầu thu phí.

- Highways: cho biết tuyến đường được tính nên tránh đường cao tốc.

- Ferries: cho thấy tuyến đường được tính toán nên tránh phà.

- Indoor: cho biết rằng tuyến đường được tính toán nên tránh các bước trong nhà cho các hướng đi bộ và chuyển tuyến. Chỉ các yêu cầu bao gồm khóa API hoặc ID ứng dụng API cao cấp của API Google Maps sẽ nhận được các bước Indoor theo mặc định.

* Language - Ngôn ngữ để trả về kết quả.

API làm hết sức mình để cung cấp địa chỉ đường phố có thể đọc được cho cả người dùng và người dân địa phương. Để đạt được mục tiêu đó, nó trả về địa chỉ đường phố bằng ngôn ngữ địa phương, được chuyển ngữ thành một tập lệnh có thể đọc được bởi người dùng nếu cần, quan sát ngôn ngữ ưa thích. Tất cả các địa chỉ khác được trả về bằng ngôn ngữ ưa thích. Tất cả các thành phần địa chỉ đều được trả về bằng cùng ngôn ngữ, được chọn từ thành phần đầu tiên.

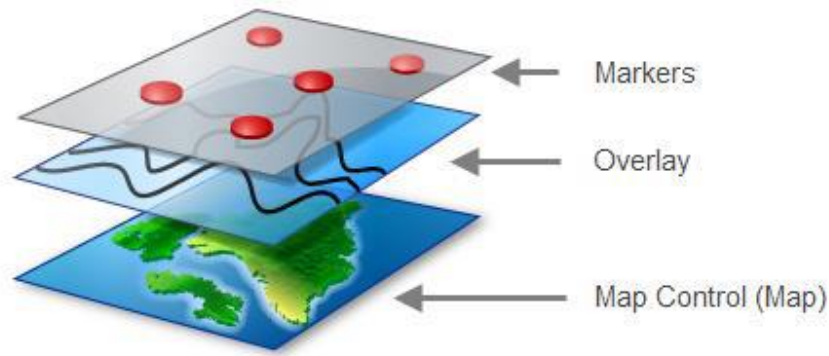
2.5.4. *Nền tảng GMap.Net*

GMap.NET là công cụ tuyệt vời và mạnh mẽ, miễn phí, chạy trên nền tảng chéo, có mã nguồn mở do .NET kiểm soát.

GMap.NET cho phép sử dụng định tuyến, mã hóa địa lý, chỉ đường và hỗ trợ các loại bản đồ từ Google, Yahoo !, Bing, OpenStreetMap, ArcGIS, Pergo, SigPac, Yandex, Mapy.cz, Maps.lt, iKarte.lv, NearMap, OviMap, CloudMade, WikiMapia, MapQuest trong Windows Forms & Presentation, hỗ trợ bộ nhớ đệm và chạy trên nền tảng của điện thoại di động!

Một vài điều cần hiểu khi sử dụng bản đồ:

- + GMapControl: là điều khiển hiển thị bản đồ
- + GmapOverlay: là một lớp trên cùng của điều khiển bản đồ. Bạn có thể có một vài lớp trên đầu bản đồ, mỗi lớp đại diện cho một tuyến đường có điểm dừng, một danh sách các cửa hàng, v.v.
- + GMapMarker: là những điểm trên một lớp, mỗi điểm đại diện cho một vị trí địa lý cụ thể (Lat, Lon) ví dụ như mỗi điểm thả trên một tuyến đường.
- + GMapRoute: là con đường hoặc hướng giữa hai hoặc nhiều điểm.



Hình 2.11. Mô hình chung của Gmap.NET

Một số thuộc tính của GmapControl:

- CanDragMap = true: người dùng có thể kéo bản đồ bằng cách sử dụng nút chuột phải.

MarkersEnabled = true: bất kỳ điểm đánh dấu nào đã xác định sẽ được hiển thị. Nếu không, chúng sẽ không xuất hiện.

- PolygonsEnabled = true: bất kì đa giác nào đã xác định sẽ được hiển thị.

- ShowTileGridLines = true: GMap.NET sẽ hiển thị tọa độ kiểu grid

- Zoom, MinZoom, MaxZoom - Mức của Zoom cho Google Maps từ 0 (thu nhỏ đến mức toàn cầu) đến 18 (phóng to đến mức đường phố). Zoom là mức thu phóng hiện tại (5 sẽ tốt cho cấp quốc gia), trong khi MinZoom và MaxZoom phải được đặt thành 0 và 18 tương ứng nếu muốn người dùng có thể phóng to và thu nhỏ hoàn toàn.

2.6. Hướng tiếp cận của đồ án

Đồ án xây dựng hệ thống giám sát camera dựa trên công nghệ nhận diện biển số xe sử dụng kỹ thuật học máy. Hệ thống gồm 3 thành phần chính:

- Phần mềm phía client sử dụng để quản lý hệ thống, giám sát hành trình xe trên bản đồ số.

- Server quản lý và điều khiển luồng dữ liệu từ các camera

- Module nhận diện biển số xe và trả kết quả về server.

Kết luận chương 2

Chương 2 đã đưa ra những lý thuyết cơ bản về các công nghệ áp dụng vào bài toán nhận diện biển số xe, các công cụ được tích hợp vào hệ thống quản lý xe ô tô dựa trên công nghệ nhận diện biển số xe. Trong chương đã trình bày về học máy

Việc lựa chọn phương pháp phù hợp rất quan trọng để đảm bảo hệ thống hoạt động một cách hiệu quả. Trong khuôn khổ đề án, học viên lựa chọn mô hình SSD và kiến trúc MobileNet làm mạng cơ sở cho phần nhận diện biển số, mô hình TPS-ResNet-BiLSTM-Attn để nhận diện ký tự. Đồng thời hệ thống cũng tích hợp Google map API và nền tảng Gmap.Net vào phần mềm để tạo các chức năng liên quan đến bản đồ số.

Chương 3

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Các thành phần, chức năng chính và các tác nhân của hệ thống

Hệ thống được xây dựng gồm 3 thành phần cao gồm:

- Phần mềm phía client sử dụng để quản lý hệ thống, giám sát hành trình xe trên bản đồ số.
- Server quản lý và điều khiển luồng dữ liệu từ các camera
- Module nhận diện biển số xe và trả kết quả về server.

3.1.1 Các thành phần và chức năng chính

Phần mềm phía client sử dụng để quản lý hệ thống, giám sát hành trình xe trên bản đồ số

- Quản lý thiết bị camera (Địa chỉ IP, Port, chuỗi kết nối, Vị trí camera , loại camera,..) :
 - Danh sách camera và trạng thái
 - Thêm mới camera
 - Sửa thông tin camera
 - Xóa camera khỏi CSDL
- Quản lý dữ liệu biển số xe :
 - Tra cứu biển số xe đã nhận diện trên camera
 - Đưa biển số xe vào danh sách theo dõi
 - Xem chi tiết thông tin từng biển số xe đã nhận diện (thời gian, vị trí, hình ảnh trích xuất từ camera)
- Quản lý dữ liệu biển số xe theo camera:
 - Tra cứu biển số xe đã nhận diện trên từng camera

- Xem chi tiết thông tin từng biển số xe đã nhận diện (thời gian, vị trí, hình ảnh trích xuất từ camera)
- Tra cứu lộ trình của xe trên bản đồ số
- Xem hình ảnh trực tiếp camera dưới dạng lưới bình thường hoặc ở chế độ nhận diện biển số

Module nhận diện biển số xe đặt trên các server có nhiệm vụ

- Mỗi module đảm nhận thu thập dữ liệu từ các camera
- Thu nhận hình ảnh từ các camera do mình quản lý sau đó xử lý và trả về server kết quả là hình ảnh và chuỗi từ các biển số xe.

Server quản lý và điều khiển luồng dữ liệu:

- Thu nhận dữ liệu từ các server chứa các module nhận dạng biển số xe và lưu trữ vào cơ sở dữ liệu
- Trả về kết quả truy vấn cơ sở dữ liệu và hình ảnh cho client.

3.1.2. Các tác nhân của hệ thống:

Dựa trên mục đích của hệ thống, ta xác định được 2 tác nhân:

- Người quản lý toàn hệ thống (Admin): Có vai trò gì quản lý chung toàn bộ hệ thống. Quản lý quyền truy cập người dùng. Thực hiện tất cả các thao tác trên phần mềm hệ thống.
- Nhân viên vận hành: Đảm nhiệm việc thực hiện tất cả các chức năng cơ bản của hệ thống.

3.2. Cấu trúc hệ thống:

3.2.1. Phân hệ quản lý camera

Xem trực tiếp hình ảnh trực tiếp truyền về từ camera theo dạng lưới hoặc xem chi tiết từng camera với các chức năng:

- Bảng điều khiển với các thao tác điều khiển camera : Pan, Tilt, Zoom,...

- Các thao tác video như: Play, Stop, Bật hiển thị với chế độ nhận diện biển số, lưu video playback trên server hoặc trên máy client.
- Hiển thị hình ảnh biển số xe được nhận diện trên camera đó.

Quản lý camera trên bản đồ số: xem các thông tin về camera, trạng thái hoạt động, giúp người dùng có cái nhìn trực quan về hệ thống camera trên bản đồ số.

Quản lý toàn diện và đầy đủ các thông tin camera như : Địa chỉ IP, các cổng kết nối, thông tin truy cập, chuỗi kết nối, vị trí camera trên bản đồ. Bao gồm các chức năng:

- Thêm một thiết bị camera với các nội dung: Tên camera, địa chỉ ip, các cổng kết nối, loại camera, thông tin đăng nhập (Username, password), chuỗi url kết nối, vị trí của camera hỗ trợ bản đồ số.
- Cập nhật thông tin thiết bị camera với các nội dung: Tên camera, địa chỉ ip, các cổng kết nối, loại camera, thông tin đăng nhập (Username, password), chuỗi url kết nối, vị trí của camera hỗ trợ bản đồ số.
- Xóa thông tin về camera

Quản lý các đoạn videoplayback: Quản lý, xem lại các đoạn video playback được lưu trữ trên server hoặc trên máy người dùng.

3.2.2. Phân hệ theo dõi phương tiện

Theo dõi hành trình của phương tiện trong một khoảng thời gian trên bản đồ số, hiển thị chi tiết lịch sử, thời gian, địa điểm theo từng camera mà phương tiện đó đã đi qua.

Tra cứu chi tiết lịch sử, thời gian, địa điểm mà phương tiện đó đã đi qua, xem lại hình ảnh được trích xuất từ video.

Tra cứu thông chi tiết lịch sử, thời gian, địa điểm các biển số xe theo từng camera, xem lại hình ảnh được trích xuất từ video.

3.2.3. Phân hệ quản lý người dùng:

Quản lý thông tin người dùng bao gồm các thông tin như tên đăng nhập, mật khẩu, vai trò với các chức năng:

- Thêm người dùng
- Cập nhật thông tin người dùng
- Xóa người dùng khỏi CSDL

Quản lý vai trò người dùng với các quyền cho từng vai trò:

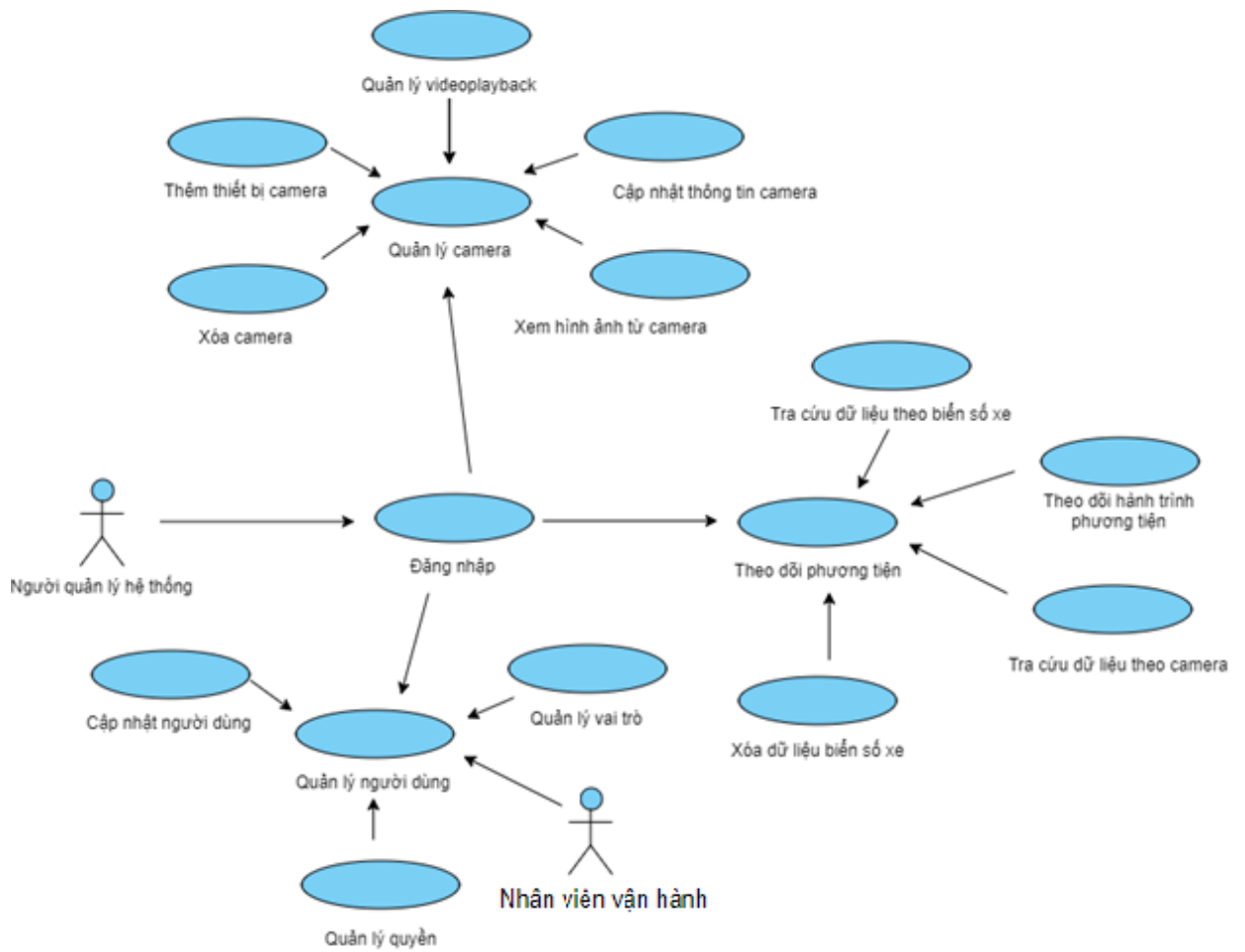
- Thêm vai trò
- Cập nhật thông tin vai trò
- Xóa vai trò người dùng

3.3. Biểu đồ Use Case

3.3.1. Các User Case của hệ thống

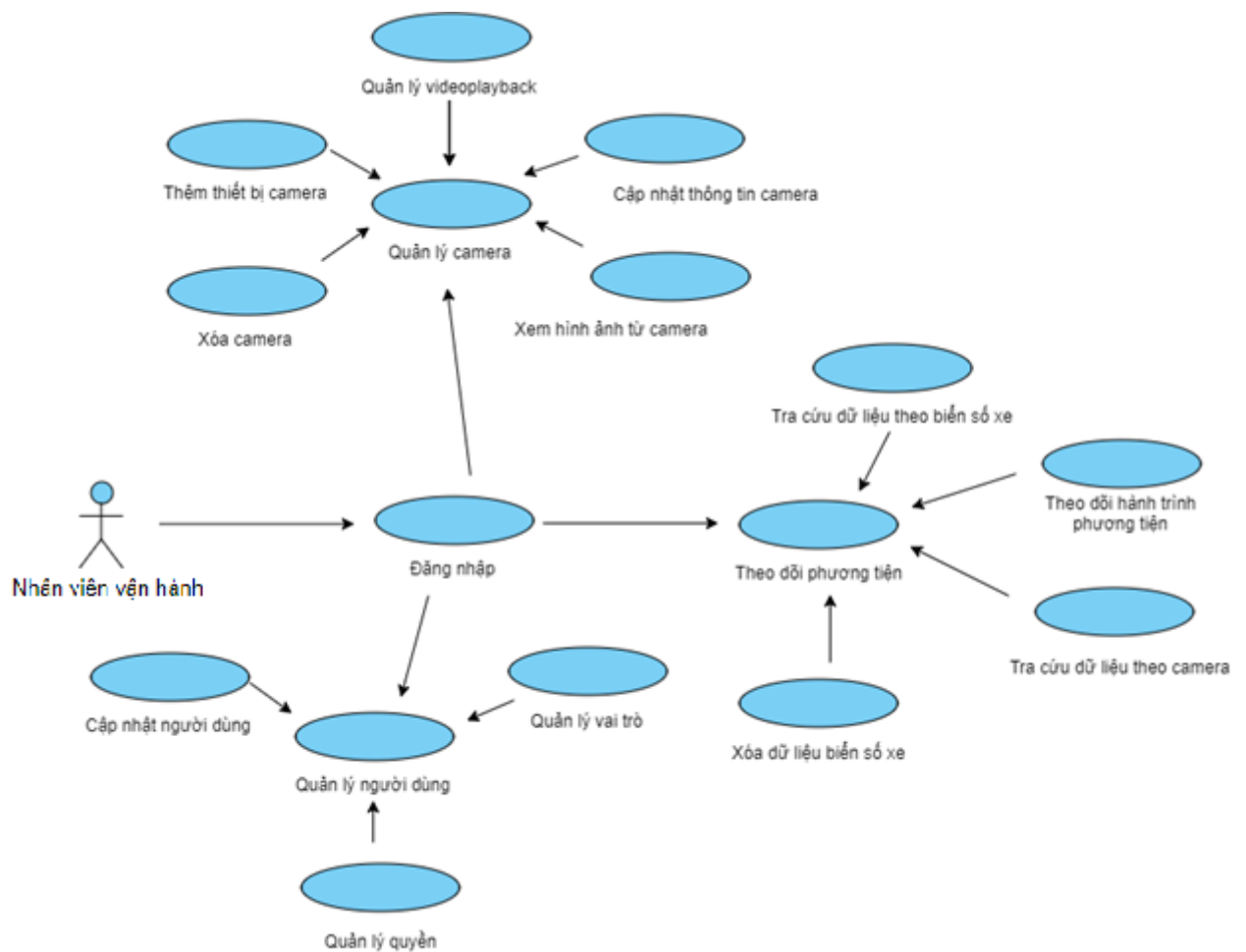
Từ 2 vai trò người dùng và usecase của hệ thống từ đó ta xây dựng các mô hình user case tương ứng với các vai trò như sau:

3.3.1.1. Mô hình use case cho vai trò người quản lý toàn bộ hệ thống



Hình 3.1. Mô hình use case cho vai trò người quản lý hệ thống

3.3.1.2. Mô hình use case cho vai trò nhân viên giám sát hệ thống



Hình 3.2. Mô hình use case cho vai trò nhân viên giám sát hệ thống

3.3.2. Đặc tả use case trong hệ thống

3.3.2.1. Use case đăng nhập

- Đối tượng sử dụng (actor) bao gồm quản lý hệ thống và nhân viên giám sát hệ thống.

- Tóm tắt: use case mô tả các đăng nhập vào hệ thống giám sát.

- Dòng sự kiện chính:

- Use case bắt đầu khi người dùng đăng nhập vào hệ thống.
- Hệ thống yêu cầu actor cung cấp thông tin đăng nhập bao gồm tên đăng nhập và mật khẩu.
- Tác nhân nhập tên đăng nhập và mật khẩu.

- Hệ thống check lại thông tin đăng nhập và thông báo thành công/thất bại cho actor. Nếu đăng nhập thành công hệ thống dựa trên thông tin đăng nhập sẽ đồng thời phân quyền tùy theo loại nhân viên. Nếu đăng nhập thất bại, hệ thống sẽ hiện thông báo cho người dùng và yêu cầu đăng nhập lại.

- Dòng sự kiện khác:

- Nhập sai tên: trong dòng sự kiện chính, nếu người dùng nhập sai tên thì hệ thống sẽ báo lỗi và yêu cầu nhập lại
- Nhập sai mật khẩu: trong dòng sự kiện chính, người dùng nhập đúng tên đăng nhập nhưng sai mật khẩu, hệ thống thông báo lỗi và yêu cầu nhập lại mật khẩu.

- Các yêu cầu đặc biệt: không có.

- Điều kiện tiên quyết: người dùng phải có tài khoản trong hệ thống

- Điều kiện kết thúc: khi người dùng đã đăng nhập được vào hệ thống.

3.3.2.2. Use case quản lý camera:

- Đối tượng sử dụng (actor) bao gồm quản lý hệ thống và nhân viên giám sát hệ thống.

- Tóm tắt: cho phép người sử dụng quản lý, theo dõi camera. Bao gồm các thao tác: Xem hình ảnh truyền trực tiếp về từ camera, xem vị trí và trạng thái camera trên bản đồ số, thao tác điều khiển camera, thêm mới một thiết bị camera vào CSDL, Lưu và quản lý video Playback, Cập nhật thông tin thiết bị camera và xóa thiết bị camera khỏi CSDL.

- Dòng sự kiện chính:

- Use case bắt đầu khi người dùng muốn xem hình ảnh truyền trực tiếp về từ camera, thao tác điều khiển camera, xem vị trí và trạng thái camera trên bản đồ số, thêm mới một thiết bị camera vào CSDL, Lưu

và quản lý video Playback, Cập nhật thông tin thiết bị camera và xóa thiết bị camera khỏi CSDL.

- Hệ thống hiển thị danh sách camera theo dạng lưới, người dùng click chọn để xem hình ảnh từ một camera với bảng điều khiển và các thao tác video, lưu video playback.
- Đối với chức năng xem vị trí camera trên bản đồ số, người dùng chọn click vào từng vị trí camera trên map để xem thông tin và trạng thái của camera
- Đối với chức năng thêm mới camera, người dùng nhập đầy đủ các trường thông tin cho camera, chọn vị trí camera trên bản đồ số, sau đó chọn thêm camera. Hệ thống sẽ kiểm tra tính hợp lệ của các thông tin, nếu thành công, camera sẽ được thêm vào CSDL.
- Đối với chức năng thêm mới camera, người dùng nhập đầy đủ các trường thông tin cho camera, chọn vị trí camera trên bản đồ số, sau đó chọn thêm camera. Hệ thống sẽ kiểm tra tính hợp lệ của các thông tin, nếu thành công, camera sẽ được thêm vào CSDL.
- Người dùng chọn xem danh sách camera, hệ thống sẽ hiển thị danh sách các camera đã có trong hệ thống, Người dùng chọn từng camera, hệ thống sẽ hiện các chức năng cho người dùng lựa chọn:
 - Nếu muốn xóa thì chọn chức năng xóa. Camera sẽ bị xóa khỏi CSDL.
 - Nếu chọn chức năng sửa, người sử dụng sẽ thay đổi các thông tin về camera. Sau khi sửa thì người sử dụng sẽ chọn chức năng cập nhật. Hệ thống sẽ kiểm tra tính hợp lệ của các thông tin, nếu thành công, thông tin camera sẽ được cập nhật vào CSDL.
- Đối với chức năng quản lý video playback người dùng chọn vị trí lưu, chọn camera, hệ thống sẽ hiển thị danh sách các video. Người dùng chọn từng video để thực hiện các thao tác:

- Nếu muốn xem video, người dùng chọn xem video.
- Nếu muốn xóa video, chọn xóa video, video sẽ bị xóa khỏi CSDL

- Dòng sự kiện khác:

- Thông tin camera không hợp lệ: khi người sử dụng nhập không đúng thì khi chọn chức năng thêm, cập nhật thì hệ thống sẽ báo lỗi và yêu cầu nhập lại.

- Điều kiện tiên quyết: người sử dụng phải đăng nhập vào hệ thống trước khi use case bắt đầu.

- Điều kiện kết thúc: khi người sử dụng đã thực hiện thành công các thao tác. Ngược lại hệ thống không thay đổi.

3.3.2.3. Use case theo dõi phương tiện

- Đối tượng sử dụng (actor) bao gồm quản lý hệ thống và nhân viên giám sát hệ thống.

- Tóm tắt: Cho phép người dùng xác định hành trình của phương tiện trên bản đồ số, tra cứu lịch sử phương tiện xuất hiện trên camera, tra cứu các biển số xe xuất hiện trên từng camera.

- Dòng sự kiện chính:

- Use case bắt đầu khi người sử dụng muốn xác định hành trình của phương tiện trên bản đồ số, tra cứu lịch sử phương tiện xuất hiện trên camera, tra cứu các biển số xe xuất hiện trên từng camera.
- Đối với chức năng theo dõi hành trình phương tiện, người dùng nhập biển số xe của phương tiện, chọn khoảng thời gian xác định sau đó chọn phân tích. Hành trình của phương tiện sẽ được hiển thị trên bản đồ số nơi có các vị trí đặt camera mà phương tiện đi qua.
- Đối với tra cứu dữ liệu theo biển số xe, người dùng nhập biển số xe của phương tiện, chọn khoảng thời gian xác định sau đó chọn phân

tích, danh sách vị trí, camera, thời gian phương tiện xuất hiện trên camera được hiển thị, người dùng chọn từng dòng thông tin để xem hình ảnh trích xuất từ camera. Người dùng có thể chọn xóa dữ liệu camera để xóa dữ liệu biển số xe đó.

- Đối với tra cứu dữ liệu theo camera, người dùng chọn camera muốn xem, chọn khoảng thời gian xác định sau đó chọn phân tích, danh sách vị trí, camera, thời gian phương tiện xuất hiện trên camera được hiển thị, người dùng chọn từng dòng thông tin để xem hình ảnh trích xuất từ camera.

- Điều kiện tiên quyết: người sử dụng phải đăng nhập vào hệ thống trước khi use case bắt đầu.

- Điều kiện kết thúc: khi người sử dụng đã thực hiện thành công các thao tác. Ngược lại hệ thống không thay đổi.

3.3.2.4. Use case quản lý người dùng:

- Đối tượng sử dụng (actor) là quản lý hệ thống.

- Tóm tắt: cho phép người sử dụng duy trì thông tin tài khoản, vai trò và quyền. Bao gồm các thao tác: xem thông tin tài khoản, thêm thông tin tài khoản, sửa thông tin tài khoản, xóa thông tin tài khoản, xem thông tin vai trò, thêm thông tin vai trò, sửa thông tin vai trò, xóa thông tin vai trò, xem thông tin quyền.

- Dòng sự kiện chính:

- Use case bắt đầu khi người sử dụng muốn xem thông tin tài khoản, thêm thông tin tài khoản, sửa thông tin tài khoản, xóa thông tin tài khoản, xem thông tin vai trò, thêm thông tin vai trò, sửa thông tin vai trò, xóa thông tin vai trò, xem thông tin quyền.
- Đối với chức năng quản lý người dùng: Hệ thống hiển thị danh sách người dùng, người dùng lựa chọn các chức năng để thực hiện

- Nếu muốn xóa thì chọn chức năng xóa. Thông tin tài khoản bị xóa khỏi hệ thống.
- Nếu chọn chức năng sửa, người sử dụng sẽ thay đổi các thông tin về tài khoản. Sau khi sửa thì người sử dụng sẽ chọn chức năng cập nhật. Hệ thống kiểm tra lại tính hợp lệ của thông tin tài khoản và cập nhập lại vào danh sách.
- Đối với chức năng quản lý vai trò: Hệ thống hiển thị danh sách vai trò, người dùng lựa chọn các chức năng để thực hiện
 - Nếu muốn xóa thì chọn chức năng xóa. Vai trò bị xóa khỏi hệ thống.
 - Nếu chọn chức năng sửa, người sử dụng sẽ thay đổi các thông tin của vai trò. Sau khi sửa thì người sử dụng sẽ chọn chức năng cập nhật. Hệ thống kiểm tra lại tính hợp lệ và cập nhập lại vào danh sách.

- Dòng sự kiện khác:

- Thông tin tài khoản không hợp lệ: khi người sử dụng nhập không đúng thì khi chọn chức năng thêm, sửa, xóa thì hệ thống sẽ báo lỗi và yêu cầu nhập lại.

- Các yêu cầu đặc biệt: chỉ có người được phân quyền thì mới có quyền thêm mới và xóa thông tin khỏi hệ thống.

- Điều kiện tiên quyết: người sử dụng phải đăng nhập vào hệ thống trước khi use case bắt đầu.

- Điều kiện kết thúc: khi người sử dụng đã thực hiện thành công các thao tác. Ngược lại hệ thống không thay đổi.

- Điểm mở rộng: hệ thống cho phép thêm tài khoản khi người sử dụng chọn chức năng thêm mới.

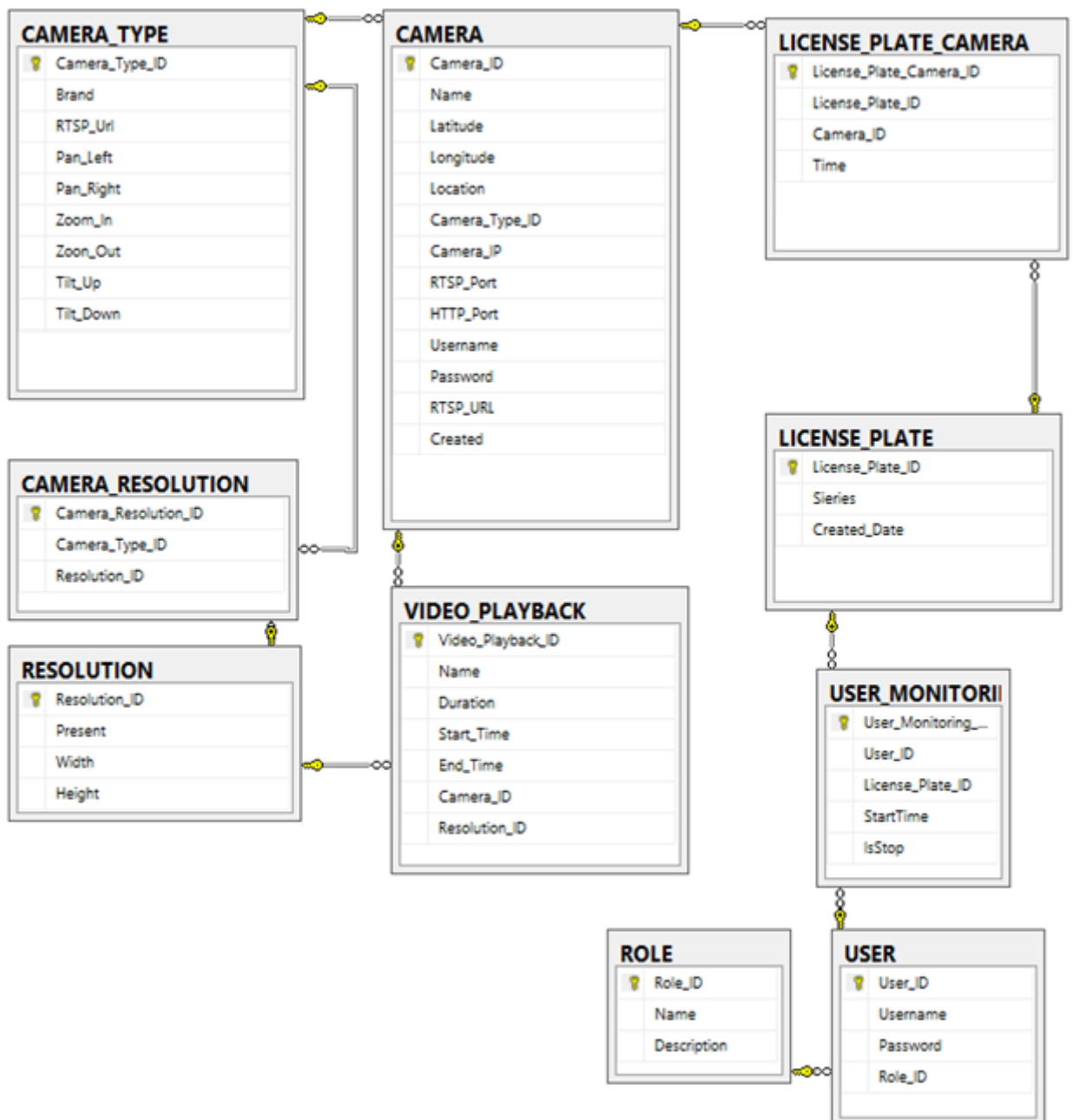
3.4. Thiết kế kiến trúc

Hệ thống sử dụng kiến trúc Client-server gồm 3 thành phần chính

- Module nhận diện biển số xe
- Server chính quản lý và xử lý dữ liệu
- Phần mềm phía client

3.5. Thiết kế dữ liệu

3.5.1. Sơ đồ logic



Hình 3.3. Sơ đồ cơ sở dữ liệu logic

3.5.2. Cơ sở dữ liệu vật lý

Bảng 3.1. Camera Type

Thuộc tính	Kiểu dữ liệu	Mô tả
Camera_Type_ID	Int	Khóa chính
Brand	Nvarchar(50)	
RTSP_Url	Ntext	
Pan_Left	Ntext	
Pan_Right	Ntext	
Zoom_In	Ntext	
Zoom_Out	Ntext	
Tilt_Up	Ntext	
Tilt_Down	Ntext	

Bảng 3.2. Camera

Thuộc tính	Kiểu dữ liệu	Mô tả
Camera_ID	Int	Khóa chính
Name	Nvarchar(200)	
Latitude	Float	

Longitude	Float	
Location	Ntext	
Camera_Type_ID	Int	
Camera_IP	Nvarchar(50)	
RTSP_Port	Nvarchar(50)	
HTTP_Port	Nvarchar(50)	
Username	Nvarchar(50)	
Password	Nvarchar(200)	
RTSP_URL	Nvarchar(200)	
Created	Datetime	

Bảng 3.3. Resolution

Thuộc tính	Kiểu dữ liệu	Mô tả
Resolution_ID	Int	Khóa chính
Present	Nvachar(50)	
Width	Int	
Height	Int	

Bảng 3.4. Camera_Resolution

Thuộc tính	Kiểu dữ liệu	Mô tả
Camera_Resolution_ID	Int	Khóa chính
Camera_Type_ID	Int	
Resolution_ID	Int	

Bảng 3.5. Video Playback

Thuộc tính	Kiểu dữ liệu	Mô tả
Video_Playback_ID	Bigint	Khóa chính
Name	Nvarchar(200)	
Duration	Decimal	
Start_Time	Datetime	
End_Time	Datetime	
Camera_ID	Int	
Resolution_ID	Int	

Bảng 3.6. License Plate

Thuộc tính	Kiểu dữ liệu	Mô tả
License_Plate_ID	Bigint	Khóa chính
Series	Nvarchar(50)	

Created_Date	Datetime	
---------------------	----------	--

Bảng 3.7. License Plate Camera

Thuộc tính	Kiểu dữ liệu	Mô tả
License_Plate_Camera_ID	Bigint	Khóa chính
License_Plate_ID	Bigint	
Camera_ID	Int	
Time	Datetime	

Bảng 3.8. User Monitoring Plate

Thuộc tính	Kiểu dữ liệu	Mô tả
User_Monitoring_Plate_ID	Bigint	Khóa chính
User_ID	Int	
License_Plate_ID	Bigint	
StartTime	Datetime	
IsStop	Bit	

Bảng 3.9. User

Thuộc tính	Kiểu dữ liệu	Mô tả
User_ID	Int	Khóa chính
Username	Nvarchar(50)	
Password	Nvarchar(50)	
Role_ID	Int	

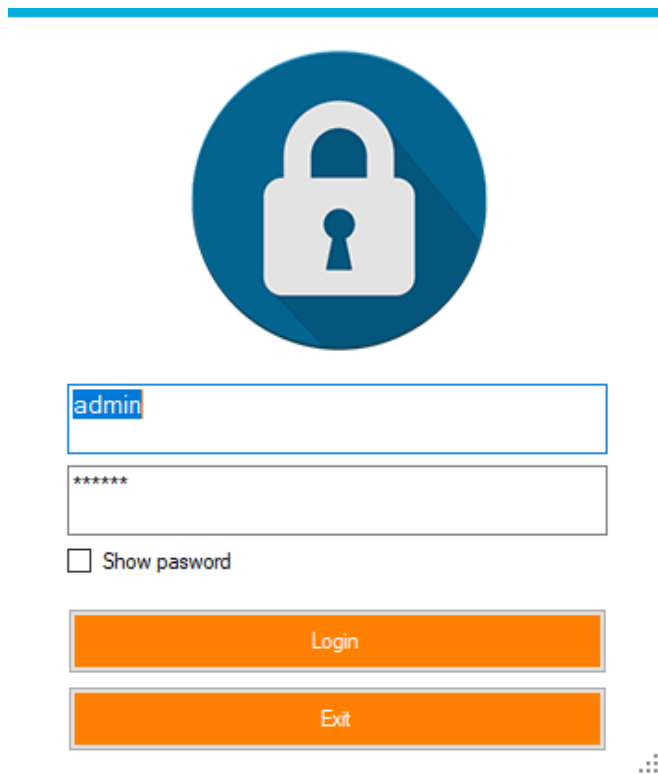
Bảng 3.10. Role

Thuộc tính	Kiểu dữ liệu	Mô tả
Role_ID	Int	Khóa chính
Name	Nvarchar(50)	
Description	Ntext	

3.6. Thiết kế giao diện

3.6.1. Giao diện đăng nhập

Giao diện đăng nhập cho phép người dùng đăng nhập vào hệ thống



admin

☐ Show password

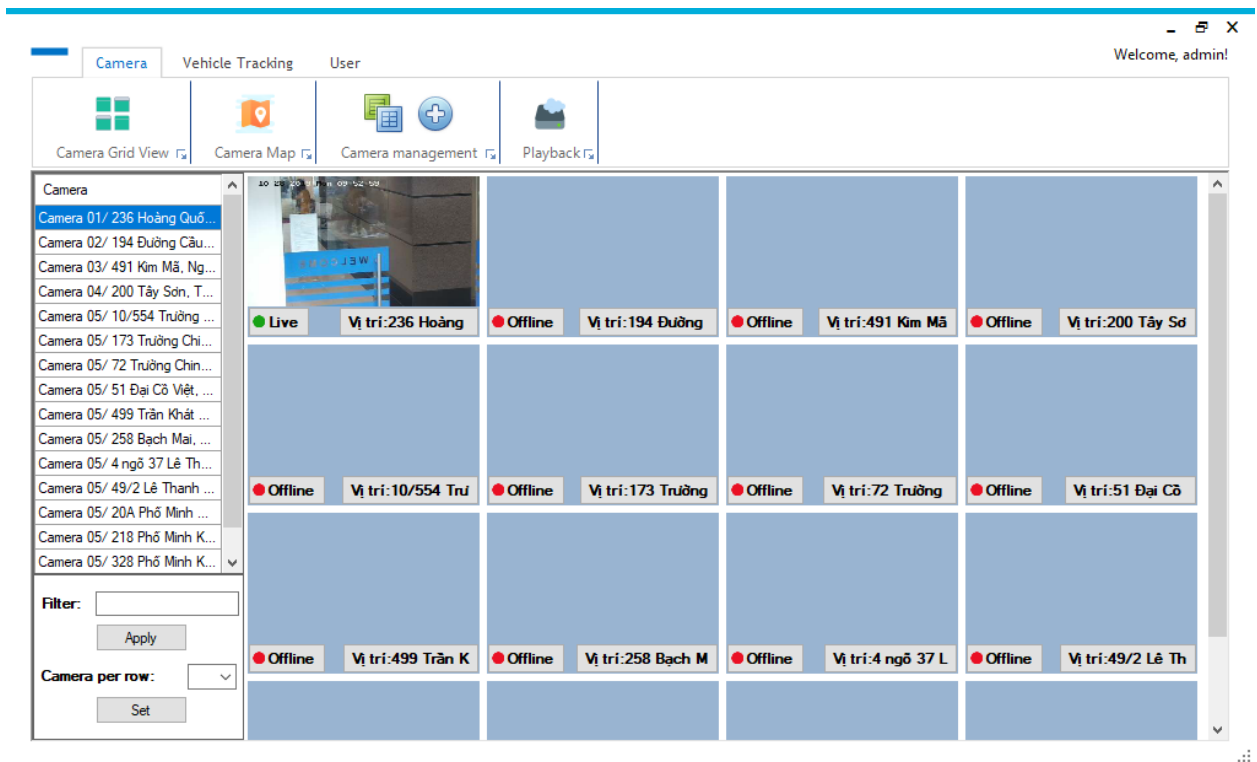
Login

Exit

Hình 3.4. Giao diện đăng nhập

3.6.2. Giao diện xem camera dạng lưới

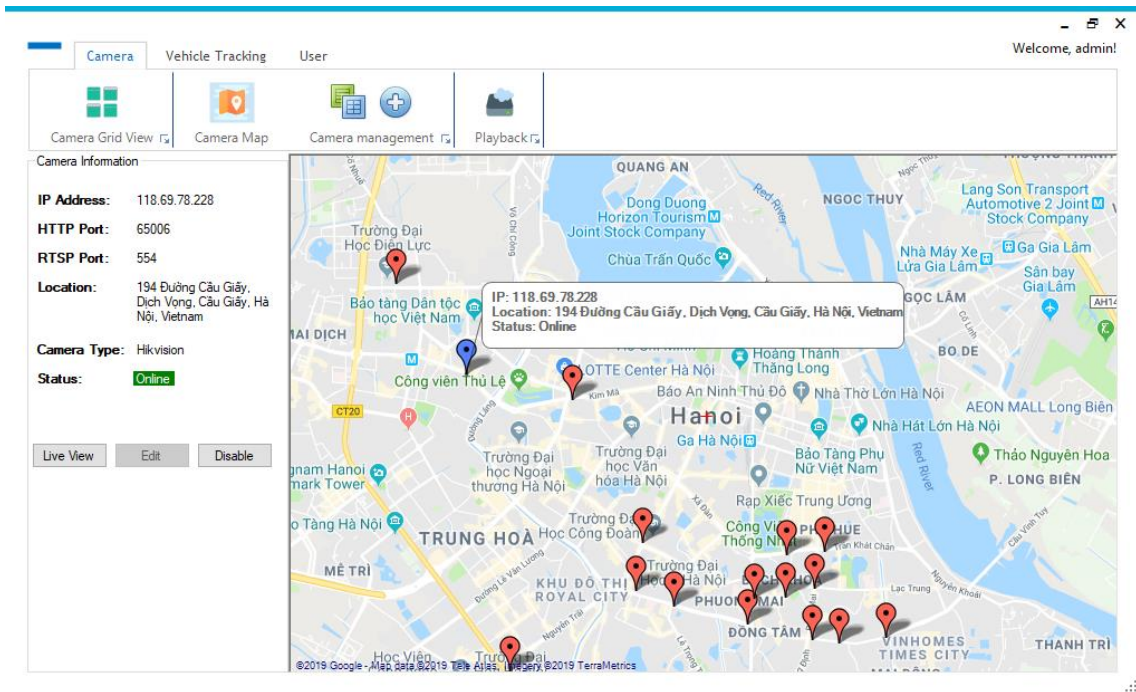
Cho phép người dùng xem các camera và chọn các camera muốn xem



Hình 3.5. Giao diện xem camera dạng lưới

3.6.3. Giao diện bản đồ camera

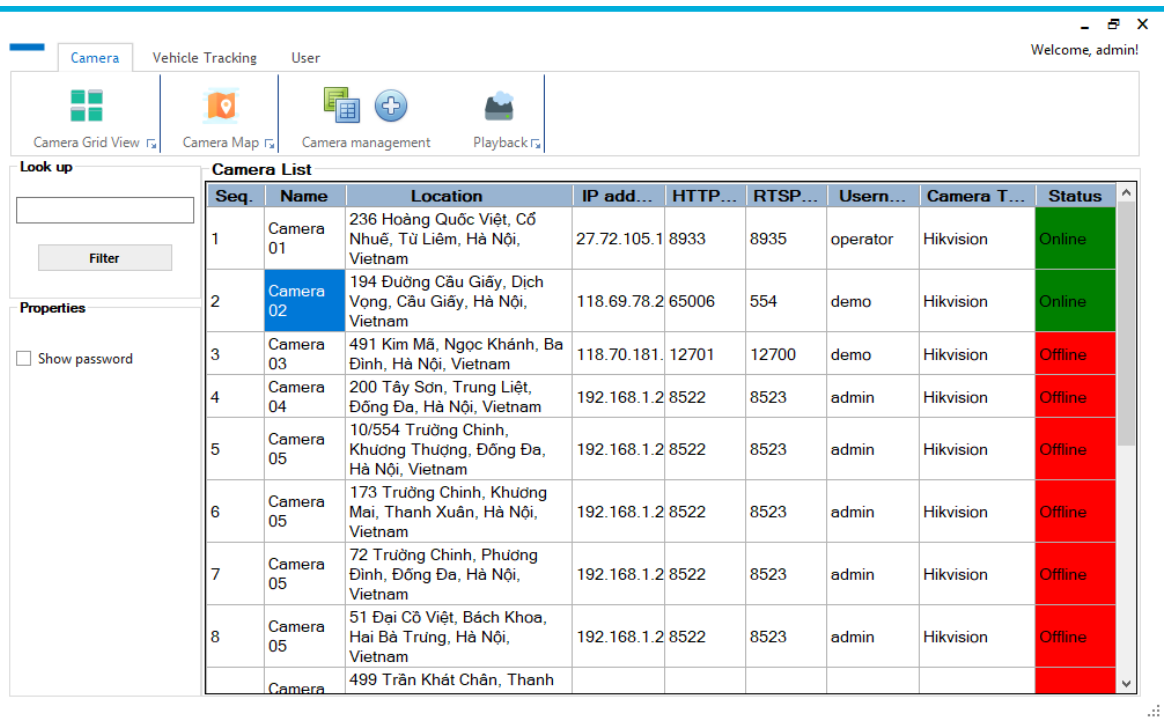
Cho người dùng có cái nhìn khái quát về vị trí và trạng thái của các camera trên bản đồ



Hình 3.6. Giao diện bản đồ camera

3.6.4. Giao diện danh sách camera

Cho phép người dùng xem thông tin các camera và trạng thái của các camera đó



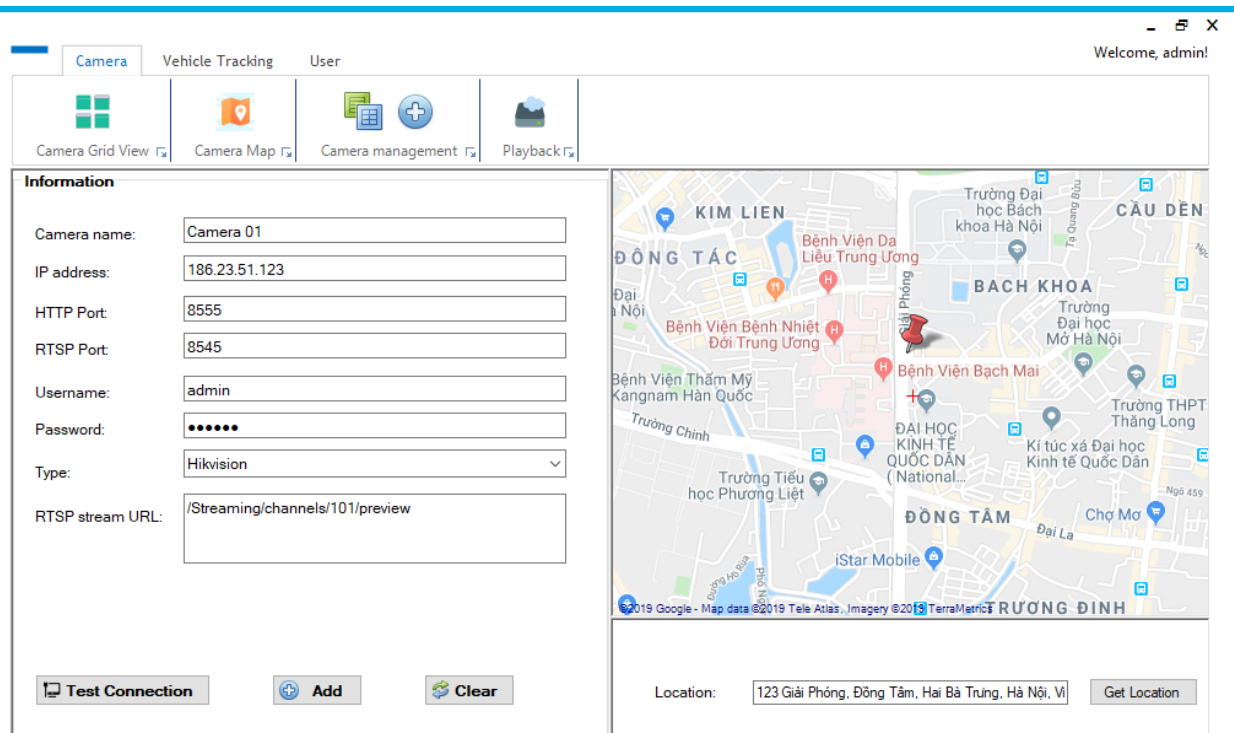
The screenshot shows a web application interface with a top navigation bar containing 'Camera', 'Vehicle Tracking', and 'User' tabs. Below the navigation bar are four icons: 'Camera Grid View', 'Camera Map', 'Camera management', and 'Playback'. The main content area is titled 'Camera List' and contains a table with the following data:

Seq.	Name	Location	IP add...	HTTP...	RTSP...	Usern...	Camera T...	Status
1	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	27.72.105.1	8933	8935	operator	Hikvision	Online
2	Camera 02	194 Đường Cầu Giấy, Dịch Vọng, Cầu Giấy, Hà Nội, Vietnam	118.69.78.2	65006	554	demo	Hikvision	Online
3	Camera 03	491 Kim Mã, Ngọc Khánh, Ba Đình, Hà Nội, Vietnam	118.70.181.	12701	12700	demo	Hikvision	Offline
4	Camera 04	200 Tây Sơn, Trung Liệt, Đống Đa, Hà Nội, Vietnam	192.168.1.2	8522	8523	admin	Hikvision	Offline
5	Camera 05	10/554 Trường Chinh, Khương Thượng, Đống Đa, Hà Nội, Vietnam	192.168.1.2	8522	8523	admin	Hikvision	Offline
6	Camera 05	173 Trường Chinh, Khương Mai, Thanh Xuân, Hà Nội, Vietnam	192.168.1.2	8522	8523	admin	Hikvision	Offline
7	Camera 05	72 Trường Chinh, Phường Đình, Đống Đa, Hà Nội, Vietnam	192.168.1.2	8522	8523	admin	Hikvision	Offline
8	Camera 05	51 Đại Cồ Việt, Bách Khoa, Hai Bà Trưng, Hà Nội, Vietnam	192.168.1.2	8522	8523	admin	Hikvision	Offline
	Camera	499 Trần Khát Chân, Thanh						

Hình 3.7. Giao diện danh sách camera

3.6.5. Giao diện thêm và cập nhật thông tin camera

Cho phép người dùng thêm mới và cập nhật thông tin các camera.



The screenshot shows a web application interface with a top navigation bar containing 'Camera', 'Vehicle Tracking', and 'User' tabs. Below the navigation bar are four icons: 'Camera Grid View', 'Camera Map', 'Camera management', and 'Playback'. The main content area is titled 'Information' and contains a form with the following fields:

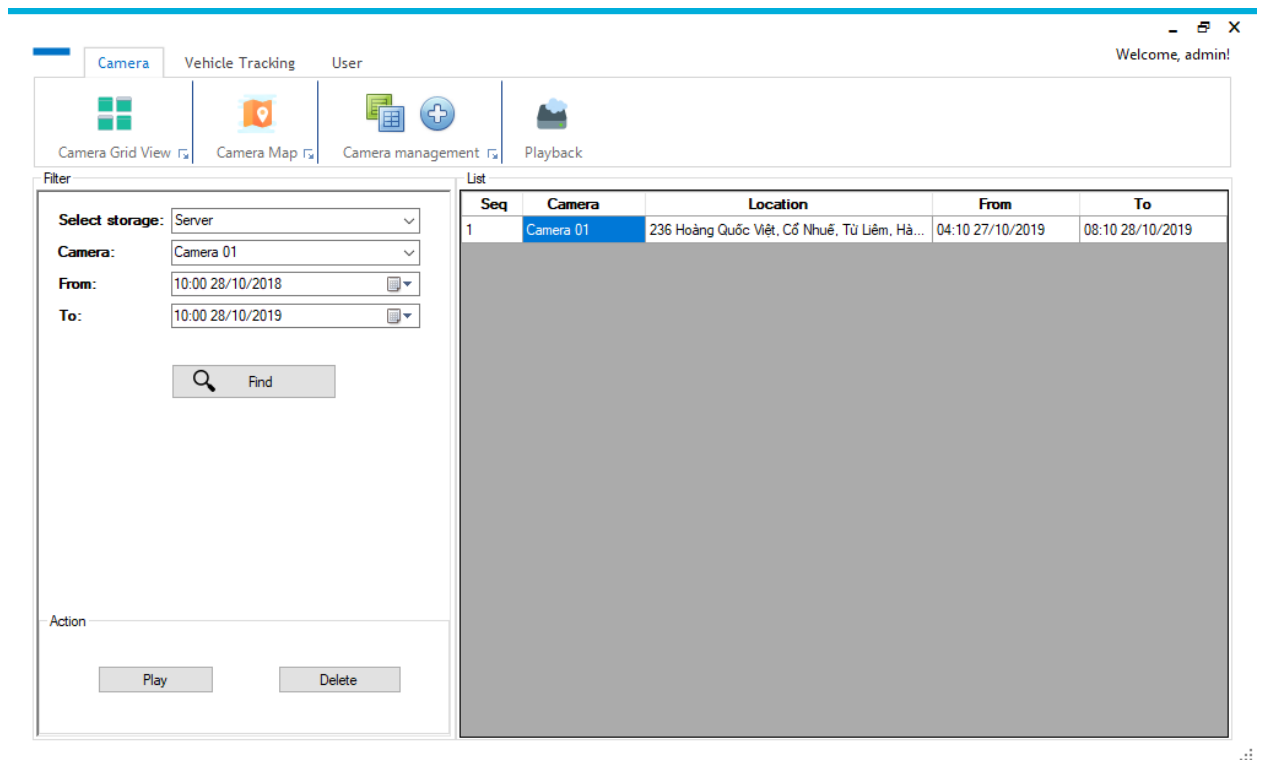
- Camera name: Camera 01
- IP address: 186.23.51.123
- HTTP Port: 8555
- RTSP Port: 8545
- Username: admin
- Password:
- Type: Hikvision
- RTSP stream URL: /Streaming/channels/101/preview

Below the form are three buttons: 'Test Connection', 'Add', and 'Clear'. To the right of the form is a map showing the location of the camera. The map is titled 'ĐÔNG TÁC' and shows various landmarks and streets. The location is marked with a red pin and labeled '123 Giải Phóng, Đông Tâm, Hai Bà Trưng, Hà Nội, Vi'.

Hình 3.8. Giao diện thêm, cập nhật camera

3.6.6. Giao diện videoplayback

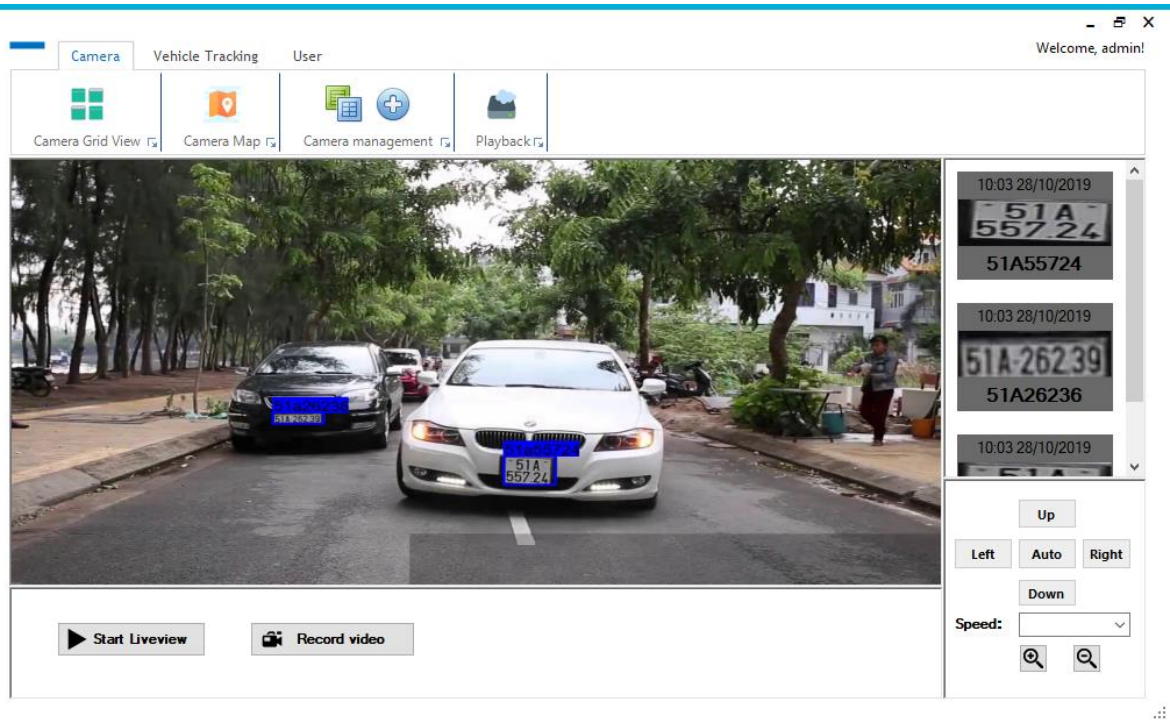
Cho phép người dùng xem và quản lý các đoạn video playback



Hình 3.9. Giao diện video playback

3.6.7. Giao diện xem camera đơn

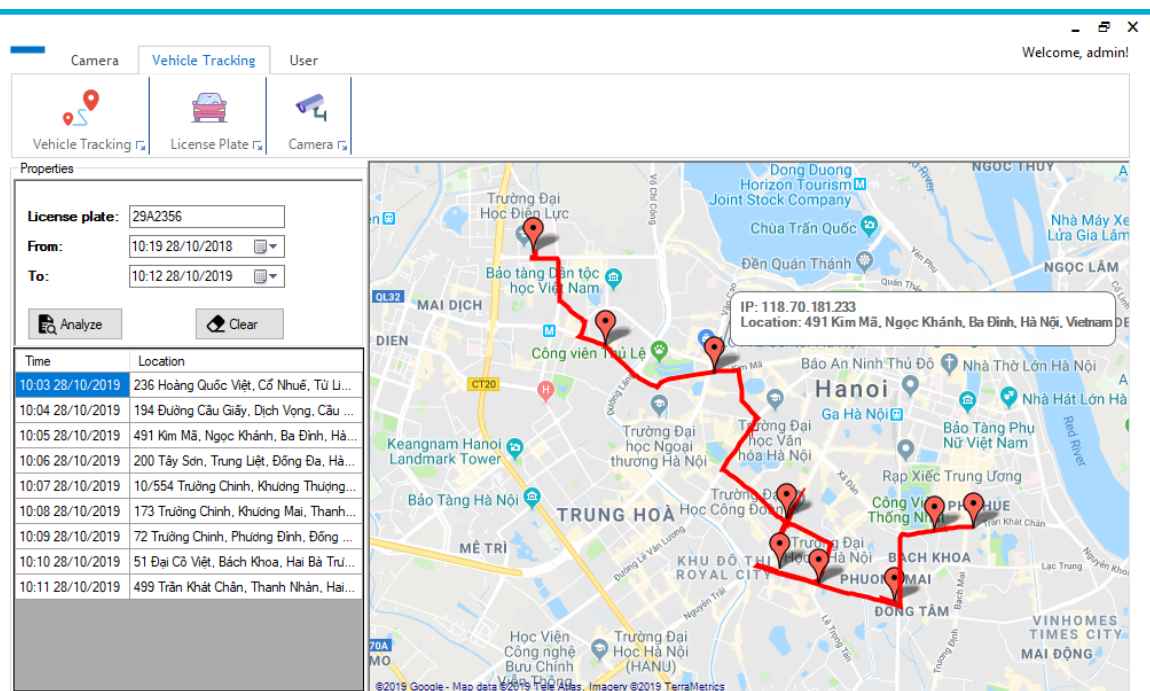
Cho phép người dùng xem camera đơn với các thao tác điều khiển ở chế độ detection



Hình 3.10. Giao diện xem camera đơn

3.6.8. Giao diện theo dõi phương tiện

Cho phép người dùng xem lộ trình của phương tiện trong một khoảng thời gian nhất định. Người dùng nhập biển số xe cần xem, chọn khoảng thời gian xác định. Sau đó chọn Analyze để hiển thị kết quả



Hình 3.11. Giao diện theo dõi hành trình

3.6.9. Giao diện xem dữ liệu nhận diện biển số xe của một biển số xe nhất định

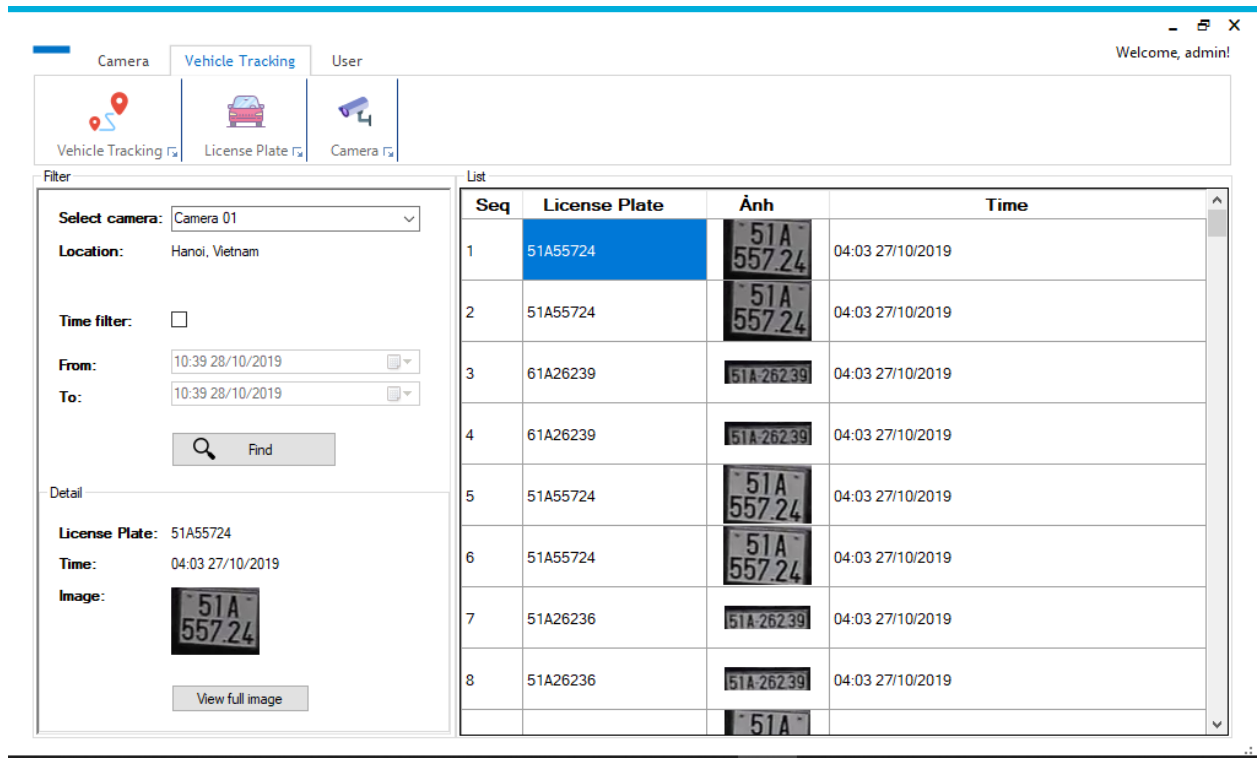
Người dùng nhập biển số xe cần xem và có tùy chọn giới hạn khoảng thời gian. Sau đó chọn Find để hiện kết quả. Người dùng chọn từng bản ghi và có thể xem ảnh đầy đủ từ camera bằng cách chọn View full image

Seq	Camera	Location	Time
1	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:25 27/10/20...
2	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:25 27/10/20...
3	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:33 27/10/20...
4	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:33 27/10/20...
5	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:41 27/10/20...
6	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:03:41 27/10/20...
7	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:03 27/10/20...
8	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:03 27/10/20...
9	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:15 27/10/20...
10	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:15 27/10/20...
11	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:24 27/10/20...
12	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:24 27/10/20...
13	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:32 27/10/20...
14	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:32 27/10/20...
15	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:58 27/10/20...
16	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:04:58 27/10/20...
17	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:05:08 27/10/20...
18	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:05:08 27/10/20...
19	Camera 01	236 Hoàng Quốc Việt, Cổ Nhuế, Từ Liêm, Hà Nội, Vietnam	04:05:16 27/10/20...

Hình 3.12. Giao diện tra cứu biển số

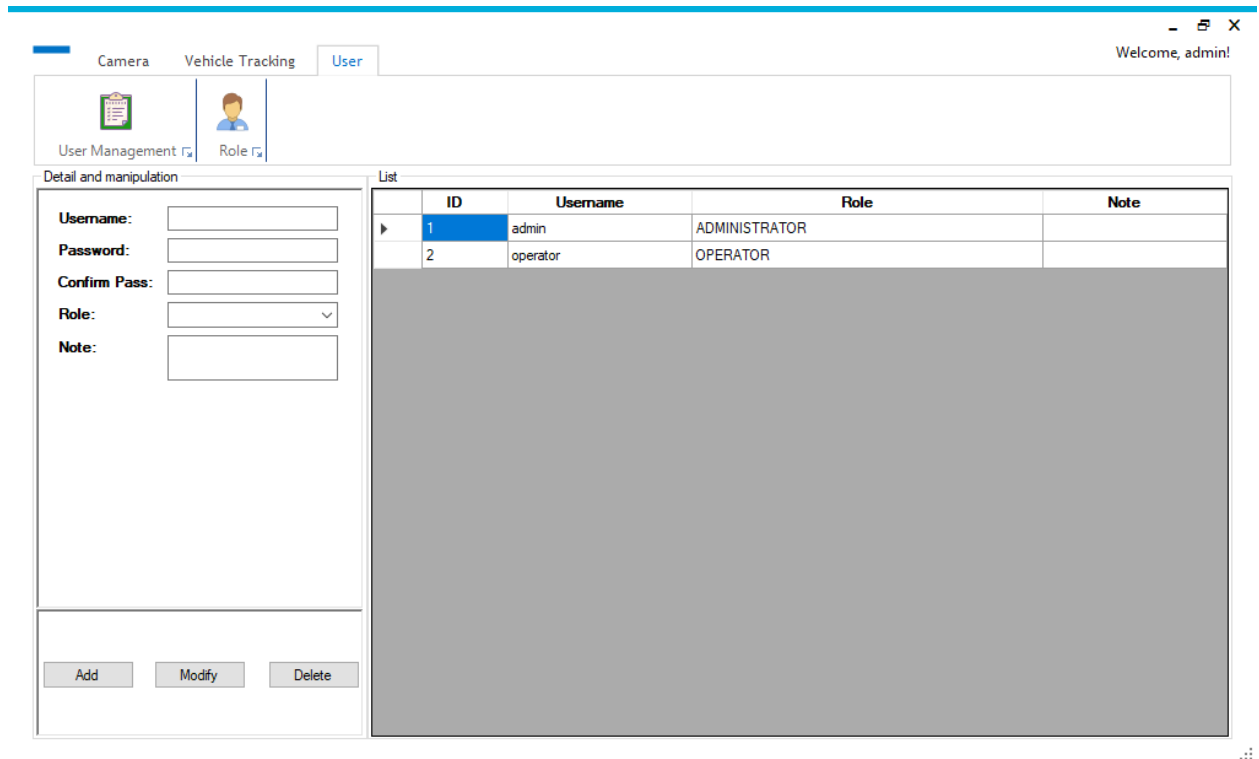
3.6.10. Giao diện xe kết quả nhận dạng biển số xe theo camera

Người dùng chọn camera cần xem và có tùy chọn giới hạn khoảng thời gian. Sau đó chọn Find để hiện kết quả. Người dùng chọn từng bản ghi và có thể xem ảnh đầy đủ từ camera bằng cách chọn View full image



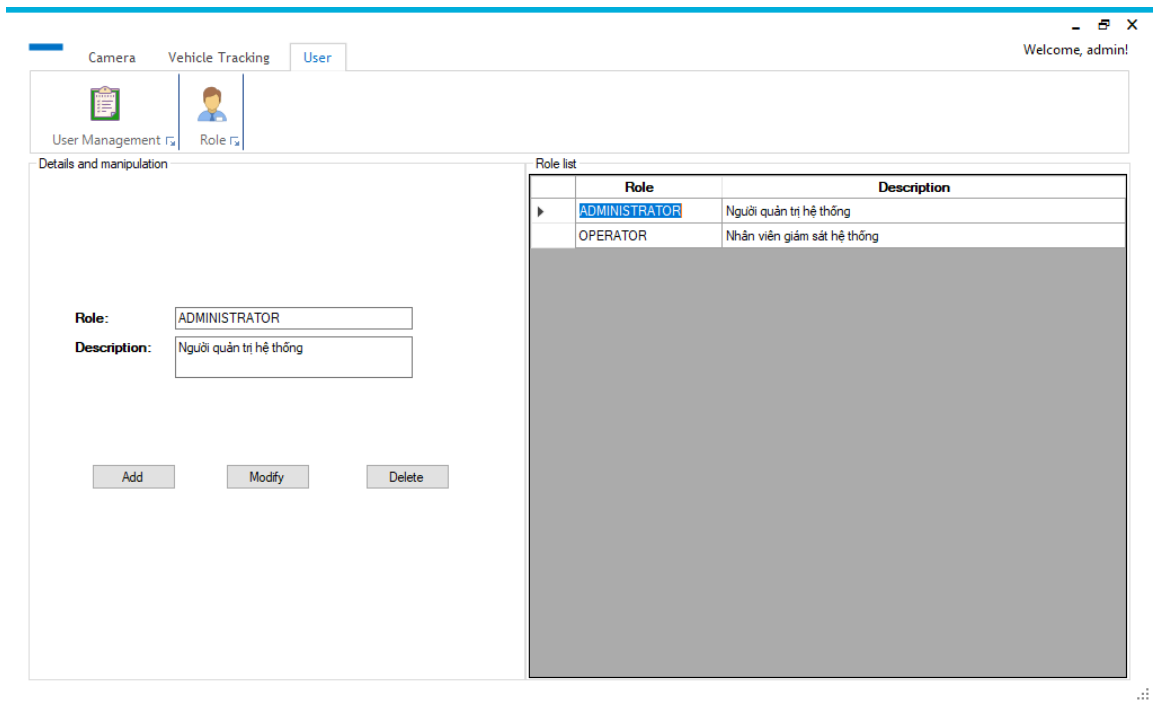
Hình 3.13. Giao diện kết quả nhận diện biển số theo camera

3.6.11. Giao diện quản lý người dùng



Hình 3.14. Giao diện quản lý người dùng

3.6.12. Giao diện quản lý vai trò



Hình 3.15. Giao diện quản lý vai trò

Kết luận chương 3

Chương 3 đã phân tích hệ thống quản lý ô tô dựa trên công nghệ nhận diện biển số xe giúp người sử dụng có thể hiểu được các chức năng chính mà hệ thống muốn xây dựng, những thông tin quan trọng, định nghĩa các yêu cầu và chức năng cụ thể. Từ những phân tích đó, trong chương 3 cũng đưa những thiết kế về cả dữ liệu và giao diện để người sử dụng có thể bắt đầu hiểu sâu hơn về hệ thống, thuận lợi trong quá trình sử dụng.

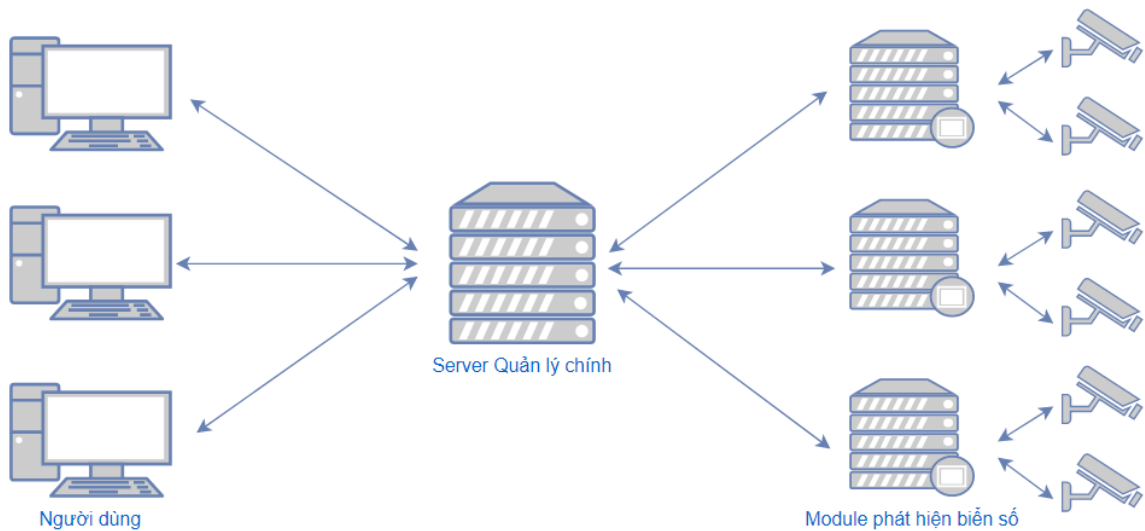
Chương 4

XÂY DỰNG HỆ THỐNG

4.1. Mô hình hệ thống

Hệ thống được chia thành 3 thành phần chính:

- Module nhận diện biển số xe
- Server chính quản lý và xử lý dữ liệu
- Phần mềm phía client



Hình 4.1. Mô hình hệ thống

4.2. Xây dựng các module

4.2.1. Module nhận diện biển số xe

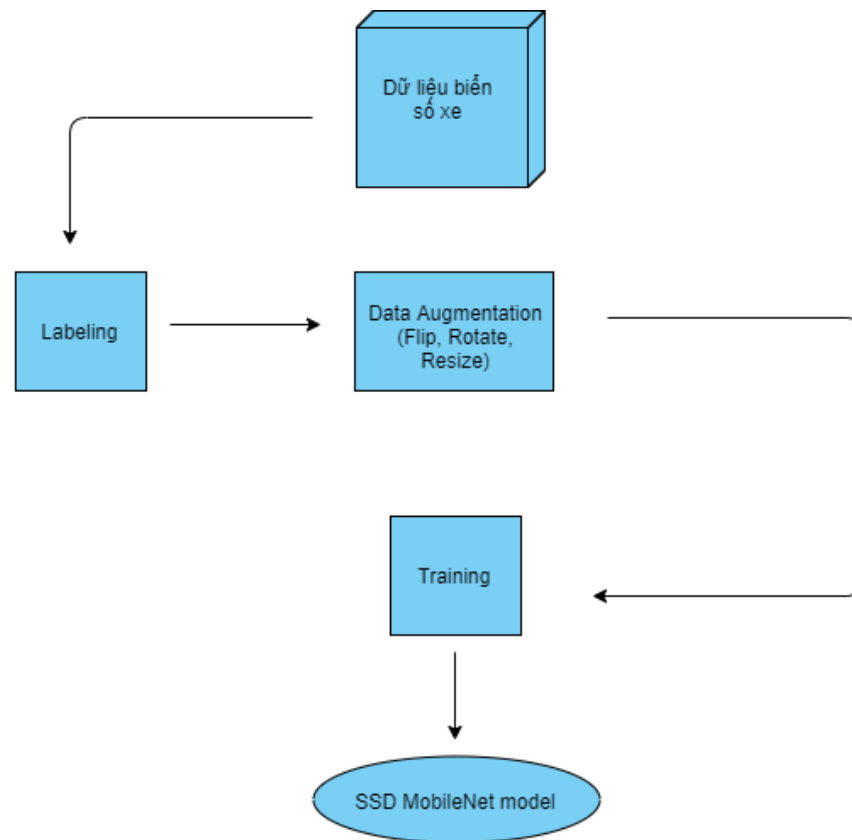
Nhận diện biển số xe chia làm hai bước:

- Bước 1: Nhận diện biển số xe trong hình ảnh trích xuất từ camera truyền về sử dụng mô hình học sâu SSD MobileNet v2.
- Bước 2: Từ biển số xe đã xử lý ở Bước 1, tiến hành nhận diện ký tự trong biển số xe sử dụng mô hình TPS-ResNet-BiLSTM-Attn.

Các biển số xe, sau khi được nhận diện, kết quả và hình ảnh trích xuất từ camera được gửi về server để lưu trữ và cập nhật vào CSDL.

4.2.1.1. Nhận diện biển số xe

a. Huấn luyện mô hình

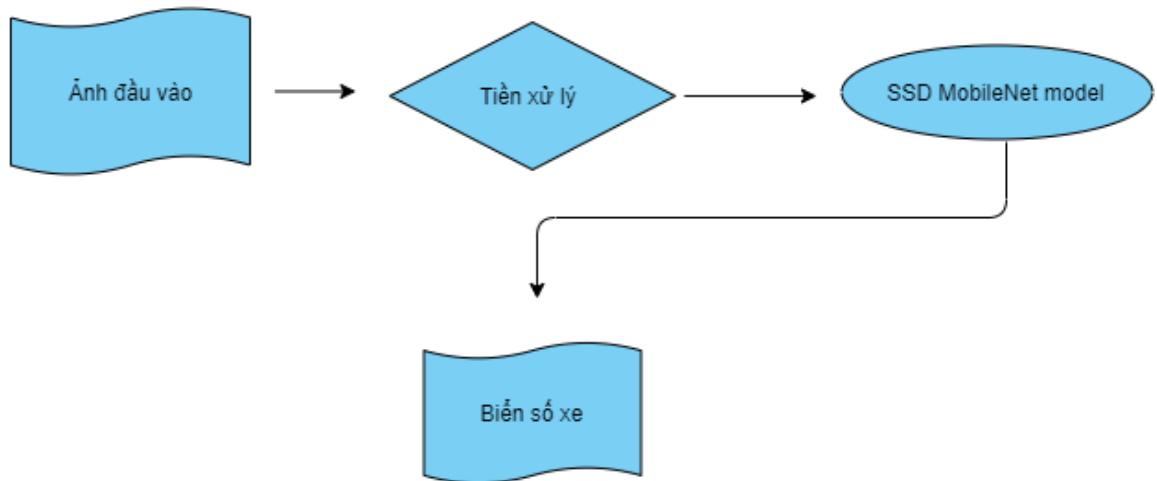


Hình 4.2. Huấn luyện mô hình nhận diện biển số xe

Dữ liệu biển số xe được gán nhãn, Sau đó tăng cường thêm dữ liệu sử dụng các phương pháp như xoay, thay đổi kích thước.

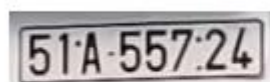
Dữ liệu sau đó được đưa vào huấn luyện để đưa ra mô hình sử dụng.

b. Sử dụng mô hình để nhận diện biển số xe



Hình 4.3. Sơ đồ nhận diện biển số xe

Kích thước các ảnh đầu vào là 300x300. Đầu ra là các ảnh biển số xe đã được phát hiện.

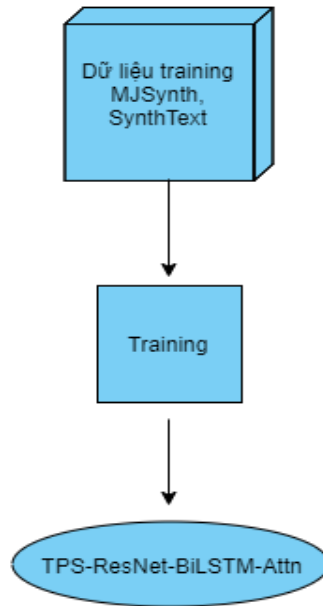


Hình 4.4. Biển số xe một dòng và hai dòng qua bước phát hiện

Ảnh đầu vào được xử lý qua các bước như thay đổi kích thước, convert về dạng GrayScale sau đó mới đưa vào mô hình để nhận diện, đầu ra ra các box chứa biển số xe.

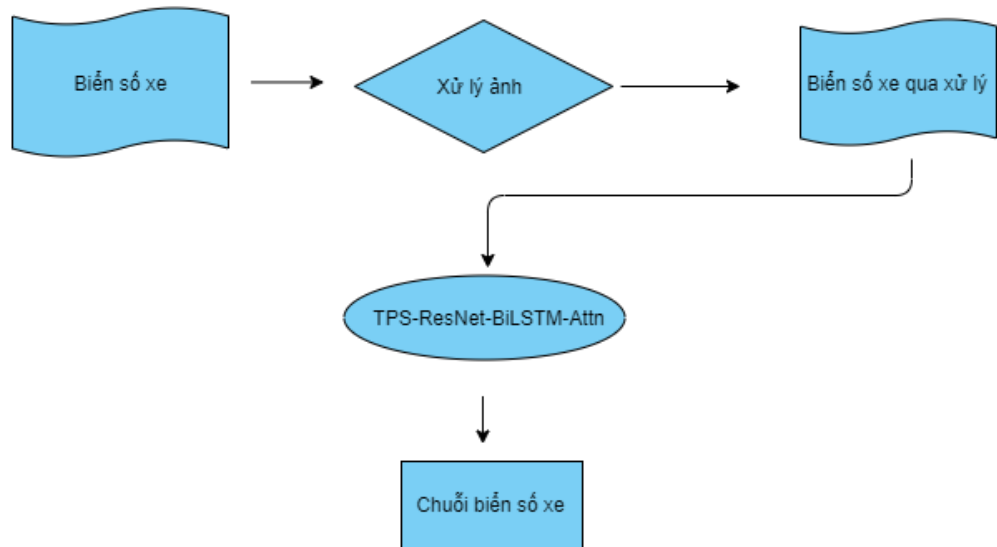
4.2.1.2 Nhận diện ký tự

a. Huấn luyện mô hình



Hình 4.5. Huấn luyện mô hình nhận diện ký tự

b. Sử dụng mô hình nhận diện ký tự



Hình 4.6. Sơ đồ nhận diện ký tự

Trước khi đưa hình ảnh biển số xe vào nhận diện, để đảm bảo kết quả được chính xác hơn, ảnh cần phải được xử lý, chỉ giữ lại phần hình ảnh trong phần khung màu đen của biển số.

Biển số xe được phát hiện ở bước 1, dùng phương pháp Harris Corner để phát hiện các điểm góc của biển số xe



Hình 4.7. Các điểm Harris Corner

Sau đó chuyển đổi phối cảnh (Perspective Transform) để đưa về khung biên số



Hình 4.8. Ảnh biển số xe sau xử lý ảnh

Ảnh đã được xử lý được đưa vào mô hình để nhận diện các ký tự. Kết quả cuối cùng là các ký tự trong biển số xe.

Đối với biển số xe có 2 dòng ký tự, phù hợp với bộ dữ liệu đã train của mô hình, chúng ta cắt biển số thành 2 phần, phần trên và phần dưới. Sau đó kết hợp 2 phần lại để được 1 biển số xe hoàn chỉnh.

4.2.2. Server chính quản lý và xử lý dữ liệu

Server chính kết nối , quản lý các server kết nối trực tiếp với các camera có vai trò :

- Kiểm tra trạng thái của camera, nếu camera ở trạng thái đang hoạt động, server sẽ tạo luồng dữ liệu riêng cho việc nhận diện biển số xe mỗi camera
- Nhận và xử lý dữ liệu gửi về từ các server quản lý camera, lưu trữ dữ liệu và quản lý CSDL.
- Trả về kết quả truy vấn cho người dùng client như lịch sử hành trình, các hình ảnh trích xuất từ camera, các video playback.

4.2.3. Phần mềm phía client

Phần mềm phía client được cài đặt trên máy người dùng với, thực hiện các chức năng cơ bản của hệ thống.

Kết luận chương 4

Chương 4 đã phân tích trình bày mô hình hệ thống quản lý xe ô tô dựa trên công nghệ nhận diện biển số xe, cấu trúc của các phần hệ thống để người dùng có thể nắm bắt được cấu trúc và cách thức hoạt động của hệ thống quản lý xe ô tô dựa trên công nghệ nhận diện biển số xe.

Chương 5

CÀI ĐẶT VÀ TRIỂN KHAI

5.1. Yêu cầu

5.1.1. Yêu cầu phần cứng

- Đối với người dùng client: Để đáp ứng được hệ thống yêu cầu tối thiểu về máy RAM tối thiểu 4G, có 10G không gian bộ nhớ và CPU 2 core trở lên. Tuy nhiên để hệ thống có thể chạy ổn định và tốt nhất thì ta cần nhiều hơn không gian bộ nhớ để cài đặt một số phần mềm liên quan. Máy tính phải có kết nối mạng Internet. Để tránh hiện tượng hoạt động chậm và sai lệch tốt nhất mạng phải ổn định có tốc độ cao (trên 24Mb/s trở lên).

- Đối với Server chính quản lý và xử lý dữ liệu: Ngoài các yêu cầu về cấu hình đối với server, tốc độ kết nối mạng cao, server cũng cần có không gian bộ nhớ lớn để lưu trữ dữ liệu từ các camera.

- Đối với module nhận diện biển số: Để đạt được tốc độ nhận diện cao, máy chủ module nhận diện biển số có cấu hình cao, trang bị card xử lý đồ họa đáp ứng các nhu cầu về tính toán, tốc độ kết nối mạng cao và ổn định

5.1.2. Yêu cầu phần mềm

- Đối với máy người dùng và server máy chủ yêu cầu chạy trên hệ điều hành window có hỗ trợ . Net framework 4.0 trở lên và Microsoft SQL Server 2014.

- Đối với module nhận diện biển số có thể chạy trên cả môi trường Window hay Linux.

5.2. Cài đặt hệ thống

5.2.1. Phần cứng

Để cài đặt hệ thống tôi sử dụng máy tính cá nhân có cấu hình phần cứng như sau:

CPU	Intel Core i5 – 2450M 2.50GHZ (4CPUs)
RAM	8GB DDR3
OS	Windows 10 Professional
Card VGA	NVIDIA Geforce 525M 1GB
Hardware	120GB SSD + 512GB HDD

Bảng 5.1. Cấu hình máy triển khai hệ thống

Để thực hiện huấn luyện mô hình nhận diện tôi sử dụng máy ảo của Google Colab.

5.2.2. Môi trường cài đặt

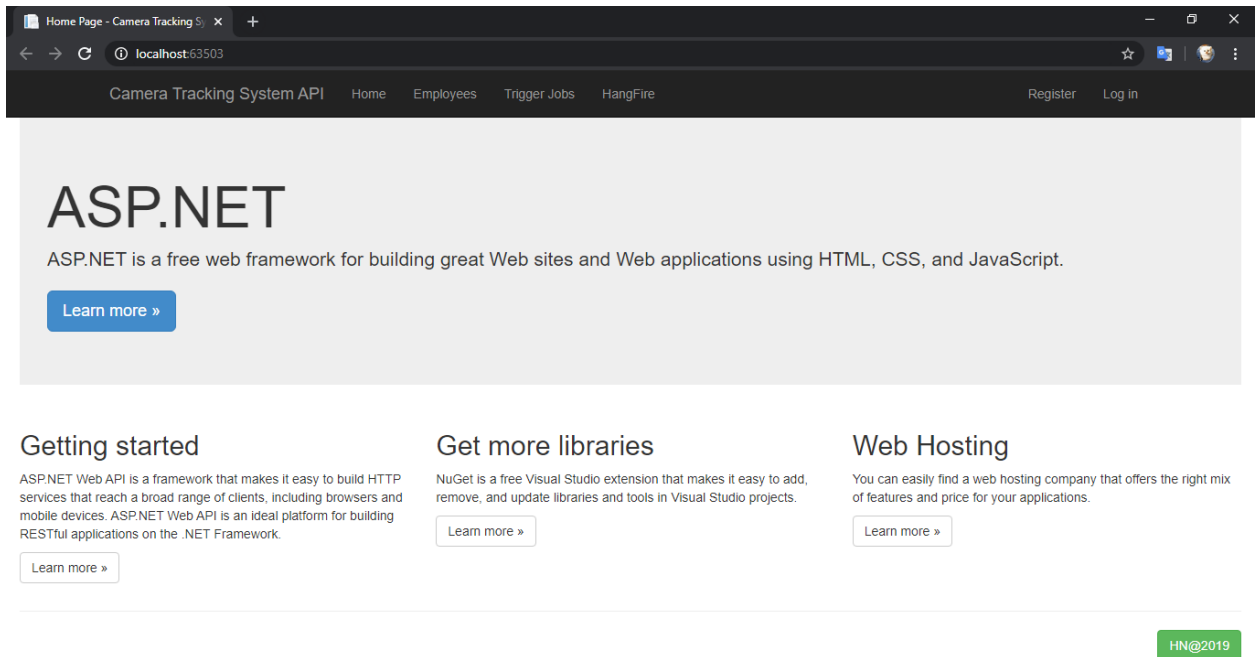
Để cài đặt được hệ thống, máy tính đã được cài đặt:

- .Net framework 4.6
- IDE: Visual Studio 2015, Visual Studio Code
- Microsoft SQL Server 2014
- Python 3.6.8
- Tensorflow-cpu 1.14.0

5.3. Thực thi phần mềm

5.3.1. Server chính quản lý và xử lý dữ liệu

Server chạy trên nền tảng ASP .NET với cơ sở dữ liệu Microsoft SQL 2014.
Server chạy trên địa chỉ <http://localhost:63503/>



Hình 5.1. Server chính quản lý và xử lý dữ liệu

5.3.2. Module nhận diện biển số xe

Module nhận diện chạy trên môi trường python, server flask, tensorflow trên địa chỉ <http://localhost:8080/>

```

C:\Windows\System32\cmd.exe - python _stream_detect.py
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\Tailieu\DATN\TensorFlow\workspace\running>python _stream_detect.py
WARNING: Logging before flag parsing goes to stderr.
W1028 15:22:10.439868 18900 deprecation.wrapper.py:119] From E:\Tailieu\DATN\TensorFlow\models\research\object_detection\
utils\label_map_util.py:132: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.

* Serving Flask app "_stream_detect" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
I1028 15:22:12.794662 18900 _internal.py:122] * Running on http://localhost:8080/ (Press CTRL+C to quit)
I1028 15:22:33.688486 32 _internal.py:122] 127.0.0.1 - - [28/Oct/2019 15:22:33] "GET /add_cam_watcher/1 HTTP/1.1" 200 -
I1028 15:22:33.819403 4284 _internal.py:122] 127.0.0.1 - - [28/Oct/2019 15:22:33] "GET /favicon.ico HTTP/1.1" 404 -
  
```

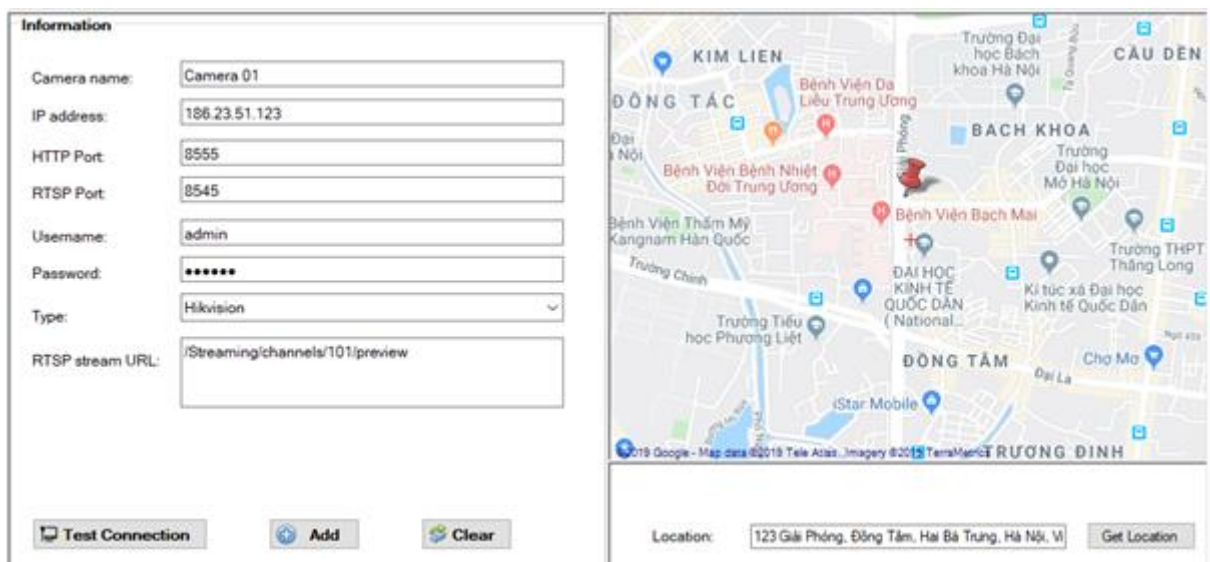
Hình 5.2. Module nhận diện biển số xe

5.3.3. Phần mềm phía client

5.3.3.1. Các chức năng sử dụng Google Map API và Gmap.NET

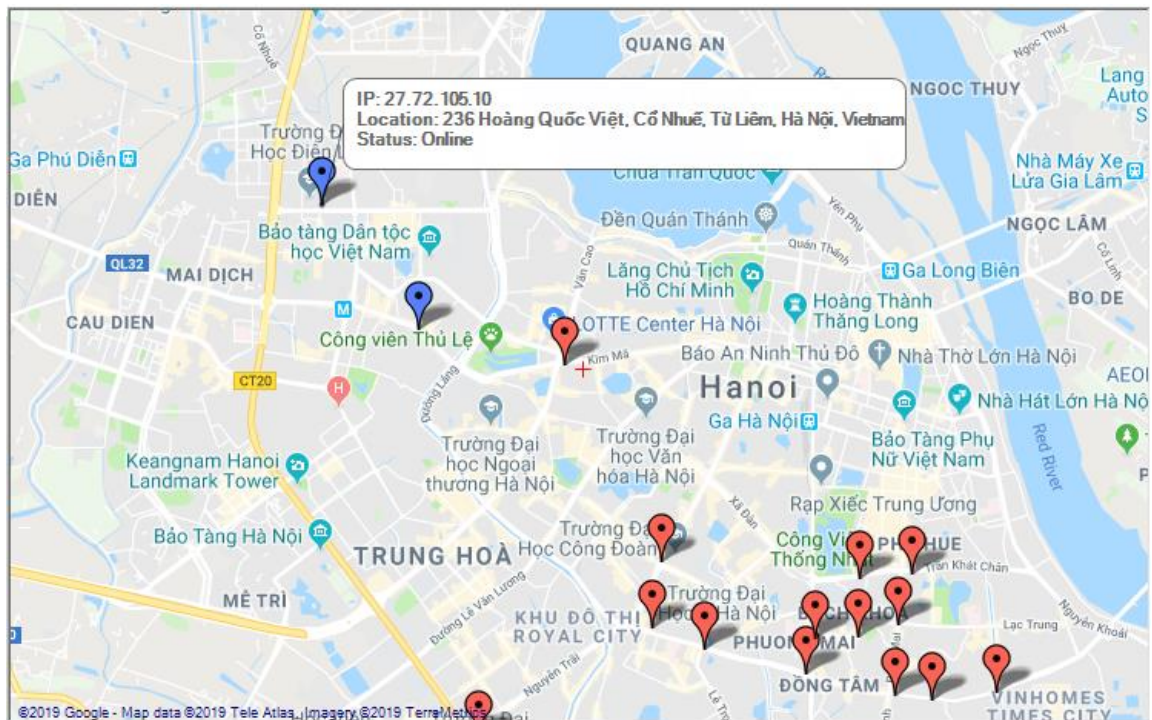
Phần mềm hệ thống quản lý xe ô tô dựa trên công nghệ nhận diện biển số xe đã ứng dụng Google Map API và GMap.Net vào một số chức năng của hệ thống như sau:

- Chức năng lấy địa chỉ khi thêm camera vào cơ sở dữ liệu. Người dùng có thể chọn vị trí camera trên bản đồ hoặc nhập địa chỉ sau đó chọn Get Location để lấy địa chỉ chính xác



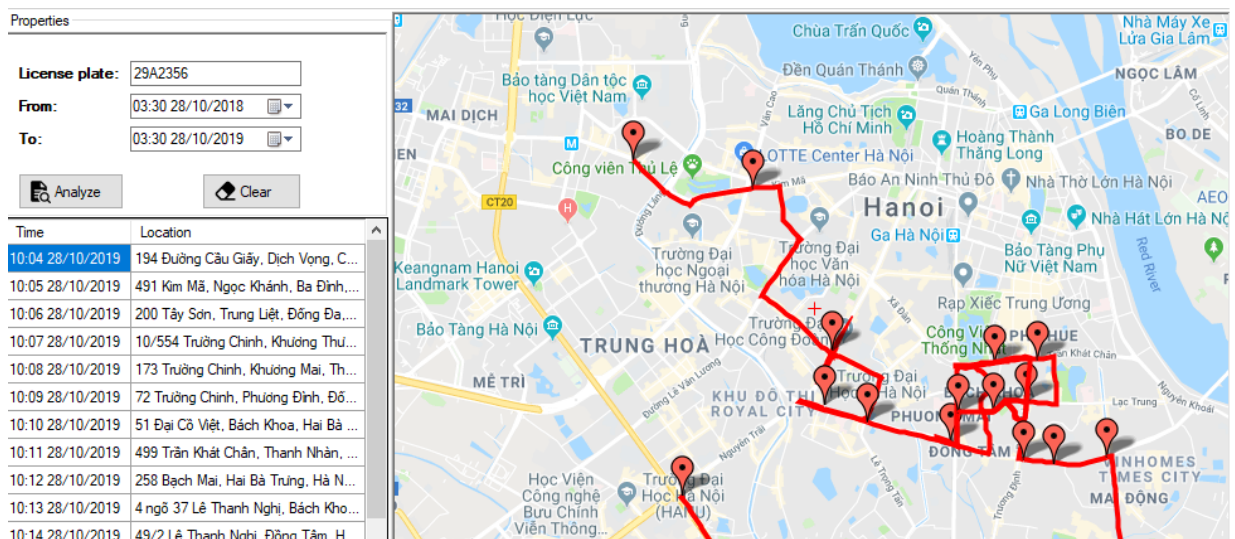
Hình 5.3. Chọn vị trí camera trên bản đồ

- Chức năng hiển thị vị trí camera trên bản đồ và trạng thái của camera. Thông qua đó, người dùng có cái nhìn trực quan về vị trí, cách bố trí của các camera. Người dùng có thể click vào các điểm đánh dấu (Marker) để hiển thị thông tin chi tiết của camera



Hình 5.4. Xem vị trí và trạng thái của các camera

- Chức năng hiển thị hành trình của phương tiện: Người dùng nhập biển số xe muốn xem hành trình, chọn khoảng thời gian thích hợp sau đó nhấn Analyze để hiển thị kết quả thứ tự xuất hiện của phương tiện trên các camera và hiển thị thành lịch trình cụ thể theo thời gian.

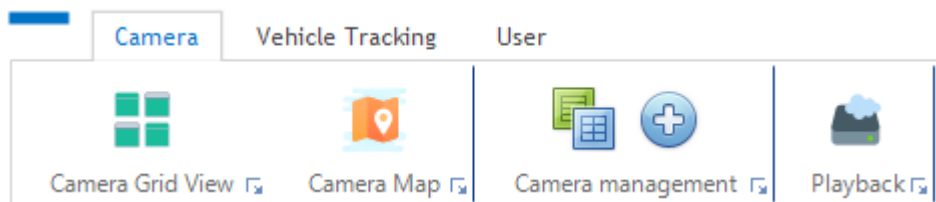


Hình 5.5. Hành trình của phương tiện trên bản đồ

5.3.3.2. Các chức năng có trên menu

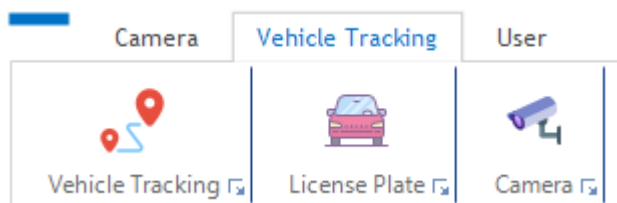
Có 3 tab chức năng chính trên phần mềm:

- Tab menu đầu tiên bao gồm các chức năng liên quan đến camera như xem camera (Camera Grid View), Hiển thị vị trí camera trên bản đồ (Camera Map), Danh sách camera, Thêm và cập nhật camera (Camera management), Quản lý các đoạn video playback (Playback).



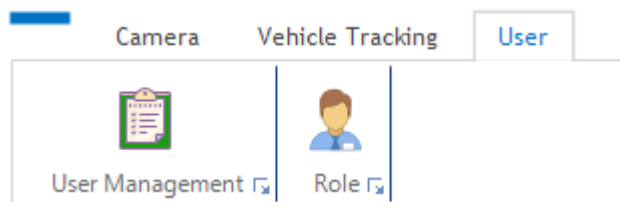
Hình 5.6. Menu chức năng camera

- Tab thứ hai bao gồm các chức năng đến quản lý biển số xe như theo dõi phương tiện (Vehicle Tracking), Tra cứu biển số xe (License Plate), Kết quả nhận diện từ các camera (Camera)



Hình 5.7. Menu chức năng theo dõi phương tiện

- Tab thứ 3 bao gồm các chức năng quản lý người dùng (User management) và Quản lý vai trò (Role).



Hình 5.8. Menu chức năng người dùng

Kết luận chương 5

Chương 5 đã trình bày toàn bộ những yêu cầu để có thể cài đặt hệ thống, hướng dẫn cài đặt và các chức năng cơ bản của hệ thống quản lý xe ô tô dựa trên công nghệ nhận diện biển số xe. Từ những nội dung đó để người sử dụng có thể sử dụng hệ thống một cách dễ dàng, hiệu quả và nhanh chóng.

KẾT LUẬN

Các kết quả đã đạt được

Đồ án đã xây dựng được mô hình hệ thống nhận quản lý ô tô trên công nghệ nhận dạng biển số, xây dựng module nhận diện biển số xe dựa trên các kỹ thuật học máy, hiển thị trực quan trên bản đồ số, xây dựng được giao diện người dùng, để người dùng dễ dàng thực hiện các thao tác hệ thống, cụ thể:

- Chương trình dễ sử dụng, đáp ứng được những nhu cầu khách quan của công việc quản lý, giám sát phương tiện dựa trên công nghệ nhận dạng biển số.
- Thiết kế chương trình cơ bản với các chức năng cơ bản đáp ứng yêu cầu của người dùng.
- Giao diện đẹp mắt thu hút người dùng, không gây nhàm chán trong quá trình sử dụng thao tác.
- Hỗ trợ các chức năng phù hợp với việc tìm kiếm thông tin và quản lý hệ thống thông tin sự cố thiên tai.
- Ứng dụng bản đồ số vào việc xây dựng hệ thống hỗ trợ người sử dụng.

Những điểm còn hạn chế của đồ án

Mặc dù đồ án đã đạt được những kết quả nhất định nhưng vẫn chưa giải quyết được một số mặt hạn chế, do khả năng và điều kiện thời gian chưa cho phép. Một số hạn chế có thể thấy như sau:

- Module nhận diện biển số xe độ chính xác còn chưa cao.
- Chưa giải quyết trọn vẹn toàn bộ các vấn đề nảy sinh, một số giải pháp vẫn chỉ còn trên ý tưởng, chưa được thực thi.

Hướng phát triển của đồ án

Trong thời gian tới, với mong muốn hoàn thiện hệ thống, học viên sẽ tiếp tục nghiên cứu các nội dung:

Nghiên cứu các mô hình nhận diện biến số để tăng độ chính xác cũng như đáp ứng nhu cầu xử lý thời gian thực của hệ thống.

Hoàn thiện, cập nhật thêm các chức năng để cải thiện trải nghiệm của người dùng, cũng như đảm bảo sử dụng hợp lý tài nguyên hệ thống, quản lý dữ liệu hợp lý.

TÀI LIỆU THAM KHẢO

- [1] Aniruddh Puranic, Deepak K. T., Umadevi V., “Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching,” *International Journal of Computer Applications*, vol. 134, no. 1, 2016.
- [2] Jitendra Sharma, Prof Khushboo Saxena, Prof Amit Sinhal, “Comparative Study of Different Techniques for License Plate Recognition,” *Journal of Advanced Computing and Communication Technologies*, vol. 1, no. 2, pp. 2-4, 2013.
- [3] Chen Chunyu, Wang Fucheng, Cheng Baozhi, Chen Xin, Zhang Chen, “Application of image processing to the vehicle license plate recognition,” in *Atlantis Press*, Paris, 2013.
- [4] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye, “Object Detection in 20 Years: A Survey,” *arxiv*, pp. 2-5, 2019.
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy,, “SSD: Single Shot MultiBox Detector,” *arXiv1512.0232*, vol. 5, no. 5, pp. 2-4, 2016.
- [6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision,” *arXiv:1704.04861v1*, vol. 1, no. 1, pp. 2-4, 2017.
- [7] “Blog on machine, Think!,” [Online]. Available: http://think4753.rssing.com/chan-74142477/all_p1.html.

- [8] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, Hwalsuk Lee, “What Is Wrong With Scene Text Recognition Model Comparisons?,” *arXiv*, pp. 4-5, 2019.