

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



LÊ NGUYỄN GIA BẢO – 18110251

TRẦN TRUNG KIÊN – 18110309

Đề Tài:

**PHÂN TÍCH TÌNH TRẠNG GIAO THÔNG
QUA THEO DÕI LUỒN LƯỢNG XE TRÊN ĐƯỜNG**

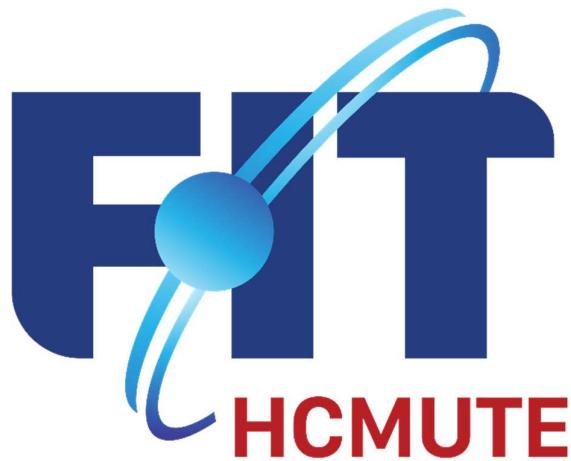
TIỂU LUẬN CHUYÊN NGÀNH KỸ SƯ CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. TRẦN NHẬT QUANG

KHÓA 2018 - 2022

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**LÊ NGUYỄN GIA BẢO – 18110251
TRẦN TRUNG KIÊN – 18110309**

Đề Tài:

**PHÂN TÍCH TÌNH TRẠNG GIAO THÔNG
QUA THEO DÕI LUỒN LƯỢNG XE TRÊN ĐƯỜNG**

TIỂU LUẬN CHUYÊN NGÀNH KỸ SƯ CNTT

**GIÁO VIÊN HƯỚNG DẪN
TS. TRẦN NHẬT QUANG**

KHÓA 2018 - 2022

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: Lê Nguyễn Gia Bảo

MSSV 1: 18110251

Họ và tên Sinh viên 2: Trần Trung Kiên

MSSV 2: 18110309

Ngành: Công Nghệ Thông Tin.

Tên đề tài: Phân Tích Tình Trạng Giao Thông Qua Theo Dõi Lưu Lượng Xe Trên Đường.

Họ và tên Giáo viên hướng dẫn: TS. Trần Nhật Quang.

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....

3. Nhược điểm:

.....
.....
.....
.....

4. Đề tài có bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

TPHCM, ngày tháng năm 2021

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: Lê Nguyễn Gia Bảo

MSSV 1: 18110251

Họ và tên Sinh viên 2: Trần Trung Kiên

MSSV 2: 18110309

Ngành: Công Nghệ Thông Tin.

Tên đề tài: Phân Tích Tình Trạng Giao Thông Qua Theo Dõi Lưu Lượng Xe Trên Đường.

Họ và tên Giáo viên hướng dẫn: TS. Trần Nhật Quang.

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....

3. Nhược điểm:

.....
.....
.....
.....

4. Đề tài có bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

TPHCM, ngày tháng năm 2021

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin phép được gửi lời cảm ơn chân thành và sâu sắc nhất đến với Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc và thực hiện đê tài này.

Bên cạnh đó nhóm chúng em xin gửi đến thầy Trần Nhật Quang lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đê tài trong thời gian qua. Thầy đã tận tâm chỉ bảo nhiệt tình nhóm chúng em trong suốt quá trình từ lúc bắt đầu cũng như kết thúc đê tài này.

Nhờ có những nền tảng kiến thức chuyên ngành vững chắc cộng thêm với những kinh nghiệm và yêu cầu thực tế ngoài xã hội thông qua việc học ở trường và thực tập ở các công ty. Tập thể các thầy cô Khoa Công Nghệ Thông Tin và đặc biệt thầy Trần Nhật Quang đã tặng cho chúng em một khối lượng kiến thức và kinh nghiệm khổng lồ về chuyên ngành và công việc trong tương lai. Đặc biệt điều này đã giúp và thôi thúc chúng em hoàn thành được đê tài. Đây sẽ là hành trang vô cùng lớn của chúng em trước khi bước ra một cuộc sống mới.

Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Chúng em hi vọng nhận được sự góp ý tận tình của quý thầy cô qua đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện nâng cấp lại sản phẩm của mình một cách tốt nhất có thể.

Nhóm chúng em xin chân thành cảm ơn!

Nhóm thực hiện

Lê Nguyễn Gia Bảo – 18110251

Trần Trung Kiên – 18110309

Trường ĐH Sư Phạm Kỹ Thuật TP.HCM

Khoa: CNTT

ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH

Họ và Tên SV thực hiện 1: Lê Nguyễn Gia Bảo Mã Số SV: 18110251
Họ và Tên SV thực hiện 2: Trần Trung Kiên Mã Số SV: 18110309
Thời gian làm luận văn: từ: 13/09/2021 Đến: 08/01/2021 (17 tuần).
Chuyên ngành: Công Nghệ Phần Mềm.
Tên luận văn: Phân Tích Tình Trạng Giao Thông Qua Theo Dõi Lưu Lượng Xe Trên Đường.
GV hướng dẫn: TS. Trần Nhật Quang.

Nhiệm Vụ Của Luận Văn:

1. Lý thuyết:

Tìm hiểu các kỹ thuật và các thuật toán về xử lý ảnh, các mô hình học máy cho bài toán nhận diện vật thể, cụ thể là nhận diện phương tiện giao thông trên đường.

2. Thực hành

- Thu thập và đánh nhãn dữ liệu thực tế từ Công Thông Tin Giao Thông Thành Phố Hồ Chí Minh.
 - Cài đặt, huấn luyện và đánh giá các mô hình cho bài toán nhận diện vật thể, cụ thể là nhận diện phương tiện giao thông trên đường.
 - Triển khai ứng dụng website.

Đề cương viết luận văn: (theo kiểu mục lục các phần, chương mục ...) (gồm các phần sau đây)

PHẦN MỞ ĐẦU	5
1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI.....	5
2. MỤC ĐÍCH NGHIÊN CỨU	5
3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	5
3.1. Đối Tượng Nghiên Cứu	5
3.2. Phạm Vi Nghiên Cứu.....	6
4. PHÂN TÍCH NHỮNG CÔNG TRÌNH CÓ LIÊN QUAN	6

5. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC	7
PHẦN NỘI DUNG.....	9
1. CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	9
1.1. Mạng Neuron Nhân Tạo	9
1.2. Mạng Neuron Tích Chập	12
1.3. Bài Toán Phân Loại Hình Ảnh	16
1.4. Bài Toán Nhận Diện Vật Thể	18
1.5. Những Ứng Dụng Khác Của Trí Tuệ Nhân Tạo Và Xử Lý Ảnh Số	22
2. CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG	24
2.1. Dữ Liệu Hình Ảnh Giao Thông Việt Nam	24
2.2. Chuẩn Hóa Và Tăng Cường Dữ Liệu	28
3. CHƯƠNG 3. NHẬN DIỆN VẬT THỂ VỚI MÔ HÌNH SSD	30
3.1. Tổng Quan Về Mô Hình SSD	30
3.2. Kiến Trúc Mạng VGG	31
3.3. Các Module Phụ Trợ.....	34
3.4. Thuật Toán Sinh Default Boxes.....	36
3.5. Huấn Luyện Mô Hình Nhận Diện Vật Thể	38
3.6. Thực Nghiệm Mô Hình SSD [18].....	43
4. CHƯƠNG 4. TRIỂN KHAI MÔ HÌNH NHẬN DIỆN PHƯƠNG TIỆN GIAO THÔNG VIỆT NAM	47
4.1. Cài Đặt Mô Hình Nhận Diện Phương Tiện Giao Thông Việt Nam	47
4.2. Quá Trình Huấn Luyện	48
4.3. Kết Quả Thực Tế Trên Tập Dữ Liệu Giao Thông Việt Nam	48
4.4. Triển Khai Website Nhận Diện Phương Tiện Giao Thông	50
4.5. Triển Khai Kênh Tin Nhắn Cảnh Báo Lưu Lượng Giao Thông	52
PHẦN KẾT LUẬN	54
1. KẾT QUẢ ĐẠT ĐƯỢC.....	54

1.1.	Kiến Thức Tìm Hiểu Được	54
1.2.	Chương Trình Thực Hiện	54
2.	ƯU ĐIỂM	54
3.	NHƯỢC ĐIỂM	54
4.	HƯỚNG PHÁT TRIỂN	55
DANH MỤC TÀI LIỆU THAM KHẢO		56
PHỤ LỤC		58

KẾ HOẠCH THỰC HIỆN

STT	Thời Gian	Công Việc	Ghi Chú
1	13/09 – 24/10	Tìm hiểu lý thuyết Thu thập dữ liệu	
2	25/10 – 14/11	Cài đặt và huấn luyện mô hình	
3	15/11 – 30/11	Thu thập dữ liệu và tiếp tục huấn luyện mô hình	
4	1/12 – 10/12	Hoàn thiện cài đặt thuật toán Triển khai ứng dụng web	
5	11/12 – 20/12	Hoàn thiện website Tổng hợp và viết báo cáo	
6	21/12 – 5/1	Hoàn thiện báo cáo	

TP. Hồ Chí Minh, ngày ... tháng ... năm 2021

Người viết đề cương

Ý kiến của giáo viên hướng dẫn

MỤC LỤC

DANH MỤC BẢNG BIỂU.....	1
DANH MỤC HÌNH ẢNH.....	2
DANH MỤC KÝ HIỆU, TỪ VIẾT TẮT.....	4
PHẦN MỞ ĐẦU	5
1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI.....	5
2. MỤC ĐÍCH NGHIÊN CỨU	5
3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	5
3.1. Đối Tượng Nghiên Cứu	5
3.2. Phạm Vi Nghiên Cứu.....	6
4. PHÂN TÍCH NHỮNG CÔNG TRÌNH CÓ LIÊN QUAN	6
5. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC	7
PHẦN NỘI DUNG.....	9
1. CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	9
1.1. Mạng Neuron Nhân Tạo	9
1.2. Mạng Neuron Tích Chập	12
1.3. Bài Toán Phân Loại Hình Ảnh	16
1.4. Bài Toán Nhận Diện Vật Thể	18
1.5. Những Ứng Dụng Khác Của Trí Tuệ Nhân Tạo Và Xử Lý Ảnh Số	22
2. CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG	24
2.1. Dữ Liệu Hình Ảnh Giao Thông Việt Nam	24
2.2. Chuẩn Hóa Và Tăng Cường Dữ Liệu	28
3. CHƯƠNG 3. NHẬN DIỆN VẬT THỂ VỚI MÔ HÌNH SSD	30
3.1. Tổng Quan Về Mô Hình SSD.....	30
3.2. Kiến Trúc Mạng VGG	31
3.3. Các Module Phụ Trợ.....	34
3.4. Thuật Toán Sinh Default Boxes.....	36

3.5.	Huấn Luyện Mô Hình Nhận Diện Vật Thê	38
3.6.	Thực Nghiệm Mô Hình SSD [18].....	43
4.	CHƯƠNG 4. TRIỀN KHAI MÔ HÌNH NHẬN DIỆN PHƯƠNG TIỆN GIAO THÔNG VIỆT NAM	47
4.1.	Cài Đặt Mô Hình Nhận Diện Phương Tiện Giao Thông Việt Nam	47
4.2.	Quá Trình Huấn Luyện	48
4.3.	Kết Quả Thực Tế Trên Tập Dữ Liệu Giao Thông Việt Nam	48
4.4.	Triển Khai Website Nhận Diện Phương Tiện Giao Thông	50
4.5.	Triển Khai Kênh Tin Nhắn Cảnh Báo Lưu Lượng Giao Thông	52
	PHẦN KẾT LUẬN	54
1.	KẾT QUẢ ĐẠT ĐƯỢC.....	54
1.1.	Kiến Thức Tìm Hiểu Được	54
1.2.	Chương Trình Thực Hiện	54
2.	ƯU ĐIỂM	54
3.	NHƯỢC ĐIỂM	54
4.	HƯỚNG PHÁT TRIỂN	55
	DANH MỤC TÀI LIỆU THAM KHẢO	56
	PHỤ LỤC	58

DANH MỤC BẢNG BIỂU

Bảng 2.1 Bảng Phân Bố Dữ Liệu Thu Thập Giai Đoạn 1	24
Bảng 2.2 Bảng Phân Bố Dữ Liệu Thu Thập Giai Đoạn 2	25
Bảng 2.3 Bảng Phân Bố Dữ Liệu Theo Lớp	26
Bảng 3.1 Bảng Giá Trị Số Lượng Default Box Mỗi Ma Trận Đặc Trung.....	38
Bảng 3.2 Các Thành Phần Ảnh Hướng Đến Hiệu Suất Mô Hình SSD [18].....	45
Bảng 3.3 Ảnh Hướng Của Số Lượng Ma Trận Trích Xuất Đặc Trung [18].....	46
Bảng 4.1 Độ Chính Xác Của Mô Hình Trên Tập Dữ Liệu Giao Thông Việt Nam	48

DANH MỤC HÌNH ẢNH

Hình 1.1 Kiến Trúc Mạng Neuron Nhân Tạo.	9
Hình 1.2 Hàm Kích Hoạt Sigmoid	10
Hình 1.3 Hàm Kích Hoạt ReLU	11
Hình 1.4 Phép Toán Tích Chập.	13
Hình 1.5 Phép Tính Tích Chập Với Bước Đệm = 1.....	14
Hình 1.6 Phép Toán Tích Chập Với Bước Nhảy = 2.	14
Hình 1.7 Phép Toán Gộp Cực Đại.....	15
Hình 1.8 Kiến Trúc Mạng LeNet.	15
Hình 1.9 Sự Phát Triển Của Mạng Neuron Tích Chập.	16
Hình 1.10 Ví Dụ Về Bài Toán Phân Loại Hình Ảnh.	17
Hình 1.11 Sơ Đồ Mô Hình Hệ Giữa Các Tác Vụ Trong Thị Giác Máy Tính.	19
Hình 1.12 Mô Hình Trích Xuất Đặc Trung Trong Bài Toán Nhận Diện Vật Thể.....	20
Hình 1.13 Mô Phỏng Giá Trị Đầu Ra Của Mạng Trích Xuất Đặc Trung.....	20
Hình 1.14 Một Số Ứng Dụng Của Trí Tuệ Nhân Tạo.....	23
Hình 2.1 Biểu Đồ Phân Bố Dữ Liệu Thu Thập Giai Đoạn 1	24
Hình 2.2 Biểu Đồ Phân Bố Dữ Liệu Thu Thập Giai Đoạn 2	25
Hình 2.3 Hình Ảnh Các Phương Tiện Giao Thông Việt Nam	26
Hình 2.4 Biểu Đồ Phân Bố Dữ Liệu Theo Lớp.....	27
Hình 2.5 Các Phép Toán Chuẩn Hóa Và Tăng Cường Dữ Liệu	29
Hình 3.1 Kiến Trúc Mô Hình SSD [18]	30
Hình 3.2 So Sánh Mạng VGG Và Mạng AlexNet.	32
Hình 3.3 Kiến Trúc Mô Hình Mạng VGG-16.....	33
Hình 3.4 Mô Phỏng Ma Trận Đặc Trung Với Tỉ Lệ Khác Nhau [18]	34
Hình 3.5 Module Xác Định Vị Trí Và Phân Loại Vật Thể	36
Hình 3.6 Minh Họa Hàm số IoU	39
Hình 3.7 Minh Họa Giá Trị IoU	39
Hình 3.8 Kết Quả Thực Nghiệm Trên Tập Dữ Liệu PASCAL VOC2007 [18].....	43
Hình 3.9 Kết Quả Thực Nghiệm Trên Tập Dữ Liệu PASCAL VOC2012 [18].....	44
Hình 4.1 Độ Chính Xác Của Mô Hình Trên Tập Dữ Liệu Giao Thông Việt Nam.....	49
Hình 4.2 Giao Diện Website Với Hình Ảnh Đầu Vào	50
Hình 4.3 Giao Diện Website Với Hình Ảnh Trực Tiếp Từ Camera Giao Thông.....	51

Hình 4.4 Giao Diện Website Với Video Đầu Vào	51
Hình 4.5 Luồng Hoạt Động Của Chức Năng Nhận Diện Từ Dữ Liệu Trực Tiếp.....	52
Hình 4.6 Hình Ảnh Kênh Tin Nhắn Cảnh Báo Lưu Lượng Giao Thông.....	53

DANH MỤC KÝ HIỆU, TỪ VIẾT TẮT

- mAP: mean Average Precision.
- R-CNN: Region Based Convolutional Neural Networks.
- SSD: Single Short MultiBox Detector.
- YOLO: You Only Look Once.

PHẦN MỞ ĐẦU

1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Với sự phát triển nhanh chóng tại các thành phố, như thành phố Hồ Chí Minh hiện tại, số lượng phương tiện tham gia giao thông ngày càng tăng. Theo số liệu từ Sở Giao thông vận tải Thành phố Hồ Chí Minh, tính đến giữa tháng 10/2020, thành phố đang quản lý hơn 8 triệu phương tiện giao thông đường bộ, trong đó có hơn 7 triệu phương tiện xe máy, và cả thành phố hiện đang còn tồn tại 22 điểm có nguy cơ ùn tắc giao thông [1]. Một số giải pháp giảm ùn tắc giao thông đã được thành phố triển khai như: tổ chức phân luồng giao thông tại các điểm có nguy cơ ùn tắc, điều chỉnh hoạt động của hệ thống đèn tín hiệu thông qua hệ thống camera giám sát giao thông.

Hiện nay, Sở Giao thông vận tải Thành phố Hồ Chí Minh đã triển khai giám sát giao thông bằng camera trên hầu hết các tuyến đường, đặc biệt là tại các giao lộ. Lượng hình ảnh thu thập từ những camera này phản ánh thực tế tình trạng giao thông tại vị trí đặt camera đó. Tận dụng lượng hình ảnh từ các camera giao thông, nhóm chúng em đã chọn thực hiện đề tài **Phân Tích Tình Trạng Giao Thông Qua Theo Dõi Lưu Lượng Xe Trên Đường**, nhằm đưa ra đánh giá về tình trạng giao thông tại các vị trí đặt camera, giúp các nhân viên quản lý sẽ dễ dàng biết được nơi nào đang kẹt xe, nơi nào đang có mật độ giao thông cao, từ đó đưa ra sự điều phối kịp thời và chính xác đến với nhân viên điều phối giao thông cũng như cảnh sát giao thông. Kết quả đánh giá trình trạng giao thông này cũng là nguồn thông tin hữu ích cho giải pháp tối ưu hoạt động của đèn tín hiệu giao thông tại các giao lộ.

2. MỤC ĐÍCH NGHIÊN CỨU

Đề tài nhằm mục đích đưa ra đánh giá mật độ tham gia giao thông thông qua hình ảnh từ các camera, nhằm giúp đưa ra cảnh báo về các khu vực đang có lưu lượng xe cao, kịp thời có giải pháp hỗ trợ giảm thiểu tình trạng kẹt xe, ùn tắc giao thông.

3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1. Đối Tượng Nghiên Cứu

Đối tượng nghiên cứu dựa trên hình ảnh từ hệ thống camera trên Cổng Thông tin Giao thông Thành phố Hồ Chí Minh [2], để đưa ra đánh giá lưu lượng giao thông hiện tại. Cụ thể như sau:

- Tập trung nghiên cứu về xử lý ảnh và bài toán Nhận diện vật thể (Object Detection), xây dựng mô hình máy học nhận diện các loại phương tiện giao thông như xe máy, xe ô tô, xe tải và xe buýt.
- Xử lý chạy đa luồng trên nhiều camera (hiện hệ thống camera giao thông của thành phố Hồ Chí Minh có khoảng hơn 600 camera, thời gian cập nhật hình ảnh của một camera khoảng 13 giây).

3.2. Phạm Vi Nghiên Cứu

Đề tài chủ yếu tập trung vào các nghiệp vụ cần thiết như: đưa ra số lượng xe mỗi loại, thông báo tình trạng giao thông, ... dựa trên hình ảnh camera giao thông Việt Nam. Phần cốt lõi là xây dựng mô hình nhận diện phương tiện giao thông với độ chính xác cao và tối ưu thời gian thực thi. Mô hình sử dụng là Single Shot Multibox Detector (SSD).

4. PHÂN TÍCH NHỮNG CÔNG TRÌNH CÓ LIÊN QUAN

Một trong những phương pháp nhận diện vật thể nổi tiếng là YOLO (You Only Look Once). Với kiến trúc YOLO, nó rất mạnh mẽ trong việc xử lý bài toán nhận diện vật thể theo thời gian thực với tốc độ khá nhanh nhưng lại hạn chế về độ chính xác, không phù đặc thù giao thông ở Việt Nam (xe cộ quá đông đúc, chủ yếu là xe máy), làm cho các phương tiện nằm sát hoặc che khuất nhau, khó nhận dạng. Bài nghiên cứu “Vehicle Detection and Counting Under Mixed Traffic Conditions in Vietnam Using YoloV4” [3] đã áp dụng phương pháp YOLO cho việc nhận diện phương tiện giao thông Việt Nam. Nhóm tác giả đã đưa ra những nhận xét về đặc tính giao thông ở các nước sử dụng xe máy là phương tiện chính như Việt Nam, cũng như những thách thức, khó khăn trong việc nhận diện phương tiện giao thông trong điều kiện trên. Nhóm tác giả tiến hành nhận diện trên năm loại phương tiện khác nhau: xe máy, xe đạp, ô tô, xe tải và xe buýt dựa trên tập dữ liệu phương tiện giao thông Việt Nam, tại hai điều kiện là thời gian gian cao điểm và thời gian thấp điểm. Nhóm tác giả công bố kết quả thực nghiệm trong cả hai điều kiện là mật độ giao thông bình thường và mật độ giao thông trong giờ cao điểm, cho thấy phương pháp YOLO có độ chính xác tốt hơn hai phương pháp khác là Haar Cascade và Background Subtraction (sử dụng phương pháp Mixture of Gaussian – MoG).

Bài nghiên cứu “Towards AI-Based Traffic Counting System with Edge Computing” [4] đã công bố nghiên cứu về việc nhận diện và theo dõi phương tiện giao

thông Việt Nam. Nhóm tác giả sử dụng phương pháp SSD và YoloV4 cho tác vụ nhận diện phương tiện, thuật toán DeepSort cho tác vụ theo dõi phương tiện. Bài nghiên cứu đã đưa ra những nhận xét về tập dữ liệu giao thông Việt Nam, phương pháp theo dõi phương tiện giao thông trên dữ liệu đường một chiều và hai chiều. Kết quả thực nghiệm đạt độ chính xác 92.1% mAP trên tập dữ liệu giao thông Việt Nam. Nhóm tác giả cũng công bố các kết quả khi triển khai mô hình trên thiết bị thực tế là bộ xử lý Nvidia Jetson Nano, đạt độ chính xác tương tự với môi trường huấn luyện trước đó trên bộ xử lý Coral Dev Board.

Bài nghiên cứu “Real-time Object Detection Based On YoloV2 For Tiny Vehicle Object” [5] đã công bố mô hình nhận diện phương tiện giao thông có kích thước nhỏ dựa trên thuật toán YoloV2, được cải tiến bằng cách thêm các khối Residual để nâng cao khả năng trích xuất đặc trưng của mô hình, nhờ đó tối ưu mô hình huấn luyện và tăng độ chính xác. Mô hình được huấn luyện trên tập dữ liệu Kitti, tập dữ liệu từ camera trước của ô tô được thu thập tại thành phố Karlsruhe, Đức. Bằng việc cải tiến mô hình YoloV2, độ chính xác được cải thiện từ 87% lên 94%, trong khi tốc độ thực thi không thay đổi, đạt được tốc độ thực thi theo thời gian thực. Bài nghiên cứu cũng chỉ ra rằng việc ảnh hưởng bởi điều kiện ánh sáng sẽ tác động đến độ chính xác của mô hình, và cần phải cải thiện trong tương lai.

Bài nghiên cứu “Detection And Classification Of Vehicles For Traffic Video Analytics” [6] đã so sánh việc nhận diện phương tiện giao thông trên hai phương pháp là Faster R-CNN và MoG kết hợp với SVM (Mixture of Gaussian and Support Vector Machine). Bài nghiên cứu đã chỉ ra sự vượt trội của mạng Convolutional Neural Network, thuật toán Faster R-CNN trong việc trích xuất đặc trưng ảnh tốt hơn, cũng như trình bày và so sánh mức độ tối ưu của thuật toán để xuất vị trí của mô hình Faster R-CNN so với hai phương pháp là R-CNN và Fast R-CNN. Kết quả huấn luyện trên tập dữ liệu “Indonesian Toll Road” cho thấy mô hình Faster R-CNN đạt kết quả tốt hơn MoG kết hợp với SVM, tuy nhiên độ chính xác của phương pháp Faster R-CNN chỉ ở mức 67.2%.

5. KẾT QUẢ ĐỰ KIẾN ĐẠT ĐƯỢC

Về kiến thức tìm hiểu được:

- Hiểu về các khái niệm, ý tưởng và các vấn đề trong việc xây dựng thuật toán và mô hình nhận diện vật thể.

- Tìm hiểu các mô hình mới và đạt hiệu suất cao trong bài toán nhận diện vật thể.

Về ứng dụng thực tế:

- Đạt được độ chính xác phù hợp và tối ưu về thời gian thực thi.
- Triển khai đa luồng trên một số lượng camera nhất định (dựa trên dữ liệu từ Công thông tin giao thông Thành phố Hồ Chí Minh).

PHẦN NỘI DUNG

1. CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

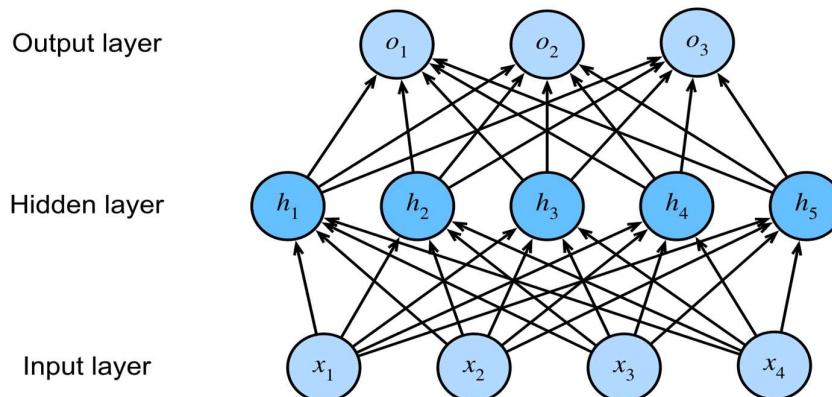
1.1. Mạng Neuron Nhân Tạo

1.1.1. Giới Thiệu Tổng Quan

Cuối thập kỷ 1940, đầu thập kỷ 1950, ý tưởng về Mạng Neuron Nhân Tạo (Artificial Neural Network) được ra đời. Mạng neuron nhân tạo được hiểu là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh của con người [7].

Mạng neuron nhân tạo có cấu trúc dựa trên các neuron (Artificial Neurons) liên kết với nhau thành các tầng. Những tầng này liên kết với nhau tạo thành một mạng neuron hoàn chỉnh. Kiến trúc mạng neuron nhân tạo được xác định bởi:

- Số lượng neuron đầu vào, số lượng neuron đầu ra.
- Số lượng tầng: mỗi mạng neuron nhân tạo gồm một tầng đầu vào (Input Layer), một tầng đầu ra (Output Layer) và một hoặc nhiều tầng ẩn (Hidden Layer).
- Số lượng neuron trong mỗi tầng.
- Cách thức các neuron trong mỗi tầng hoặc giữa các tầng liên kết với nhau.



Hình 1.1 Kiến Trúc Mạng Neuron Nhân Tạo.

(Nguồn: <https://d2l.ai>)

1.1.2. Hàm Kích Hoạt

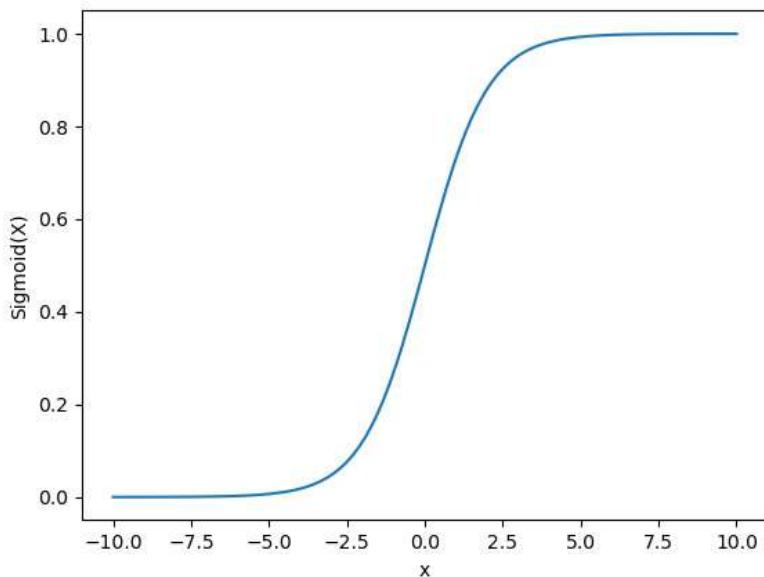
Trong mạng neuron nhân tạo, hàm kích hoạt (Activation Function) của một neuron định nghĩa đầu ra của neuron đó. Hàm kích hoạt đóng vai trò là thành phần phi tuyến của giá trị đầu vào.

Một số hàm kích hoạt tiêu biểu như:

- Hàm kích hoạt Sigmoid [8]

$$f(x) = \frac{1}{1 + e^{-x}}$$

Giá trị đầu ra của hàm số sigmoid trong khoảng (0, 1). Với giá trị đầu vào nhỏ (rất âm) hàm số sẽ trả về giá trị gần với 0, ngược lại nếu giá trị đầu vào lớn, hàm số sẽ trả về giá trị gần với 1. Hàm số này được sử dụng nhiều trong quá khứ vì có giá trị đạo hàm tại mọi điểm. Những năm gần đây, hàm số này ít khi được sử dụng, bởi sự ra đời của những hàm kích hoạt có hiệu suất tốt hơn. Hàm kích hoạt Sigmoid có một nhược điểm, đó là suy giảm đạo hàm (Sigmoid saturate and kill gradients). Khi giá trị đầu vào có giá trị tuyệt đối rất lớn (rất âm hoặc rất dương), giá trị đạo hàm tại điểm đó sẽ gần bằng 0. Khi đó, việc cập nhật các hệ số tại giá trị đầu vào này gần như không được thực hiện.



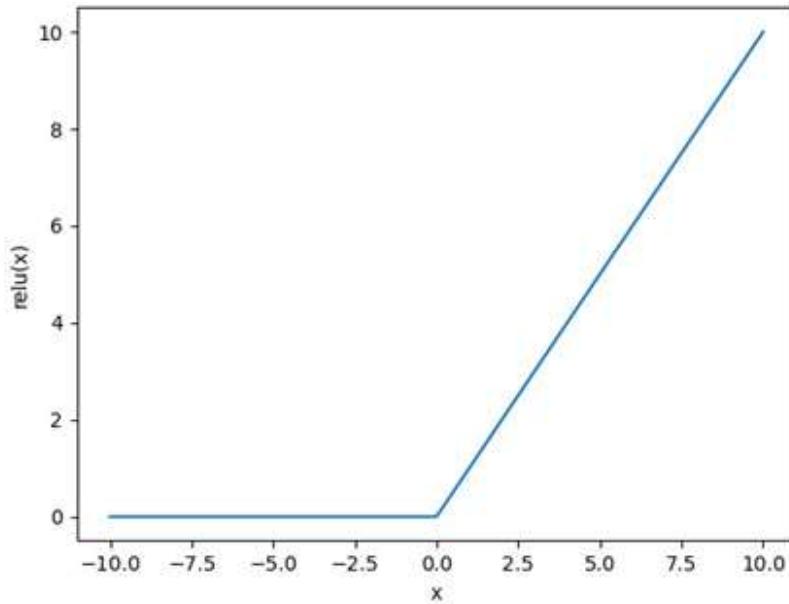
Hình 1.2 Hàm Kích Hoạt Sigmoid

- Hàm kích hoạt ReLU [9]

$$f(x) = \max(0, x)$$

Hàm kích hoạt ReLU giúp lọc giá trị đầu ra nhỏ hơn 0. Giá trị đầu ra trong khoảng [0, x]. Hàm ReLU đơn giản trong tính toán, vì vậy tối ưu trong quá trình huấn luyện. Thực nghiệm đã chứng minh rằng hàm số ReLU tối ưu quá trình hội tụ của mạng neuron nhân tạo.

Hàm kích hoạt ReLU không có đạo hàm tại $x = 0$. Trong quá trình thực nghiệm, các nhà nghiên cứu đã chứng minh được đạo hàm của hàm số ReLU tại $x = 0$ là $ReLU'(0) = 0$ và cũng khẳng định thêm rằng, xác suất giá trị đầu vào bằng 0 là rất nhỏ [10].



Hình 1.3 Hàm Kích Hoạt ReLU

1.1.3. Quy Tắc Cập Nhật Trọng Số Của Mạng Neuron Nhân Tạo

Quy tắc cập nhật trọng số được thực hiện theo các bước sau đây:

- Lan truyền tiến – Forward Propagation

Lan truyền tiến là quá trình từ một giá trị đầu vào, thực hiện các phép tính toán và hàm kích hoạt trên các tầng ẩn, trả về kết quả tại tầng đầu ra. Giá trị này được xem như giá trị dự đoán của giá trị đầu vào.

- Hàm số tính toán lỗi – Loss Function

Hàm số tính toán lỗi đơn giản có thể nghĩ là hàm đếm số lượng các giá trị đầu ra bị phân lớp lỗi (misclassified), Khi một giá trị đầu vào bị phân lớp lỗi, hàm lỗi sẽ được tính toán. Trong thực tế hàm lỗi là phép tính sai số giữa giá trị dự đoán và giá trị thực tế của một giá trị đầu vào.

Xét tập giá trị đầu vào M , các giá trị đầu vào $x_i \in M$ và giá trị đầu ra tương ứng y_i , giá trị trọng số hiện tại của mạng neuron w , ta có hàm lỗi $J(w)$ được tính bằng:

$$J(w) = \sum_{x_i \in M} (-y_i w^T x_i)$$

- Lan truyền ngược – Backward Propagation và cập nhật trọng số

Mục tiêu của việc cập nhật trọng số nhằm cực tiểu giá trị hàm lỗi $J(w)$. Khi $J(w) = 0$, ta không còn điểm nào bị phân lớp lỗi.

Việc tìm được giá trị trọng số w để giảm cực tiểu giá trị lỗi $J(w)$ được thực hiện bằng giải phương trình đạo hàm $J'(w) = 0$. Thực tế việc giải phương trình trên không khả thi trên mạng neuron. Cách tiếp cận hay được thực hiện là Suy giảm đạo hàm (Gradient Descent). Suy giảm đạo hàm là thuật toán tìm được điểm cực tiểu (cục bộ hoặc toàn cục) đạo hàm nhờ vào việc cập nhật trọng số theo ngược hướng với giá trị đạo hàm tại điểm đó.

Xét 2 điểm x_t và x_{t+1} , giá trị đạo hàm của hàm lỗi tại x_t là $J'(x_t)$, ta có

$$x_{t+1} = x_t - \eta J'(x_t)$$

Trong đó η (eta) là một số dương, được gọi là tốc độ học (learning rate). Dấu trừ trong phép tính thể hiện cho việc cập nhật giá trị mới theo hướng ngược lại của đạo hàm.

Nhiều biến thể của thuật toán Gradient Descent có hiệu quả vượt trội, giúp tối ưu quá trình huấn luyện như Stochastic Gradient Descent, hay các thuật toán tối ưu tốc độ học như Adam hay RMSprop.

1.2. Mạng Neuron Tích Chập

1.2.1. Giới Thiệu Tổng Quan

Với sự phát triển của các thuật toán xử lý ảnh và nhánh Học Sâu, Mạng Neuron Tích Chập (Convolutional Neural Network) được ra đời và phát triển như một phương pháp hiệu quả trong phân tích và nhận dạng hình ảnh. Ngày nay, mạng neuron tích chập ứng dụng trong nhiều lĩnh vực khác như xử lý ngôn ngữ hay xử lý âm thanh hay phân tích dữ liệu chuỗi thời gian (Time Series Data).

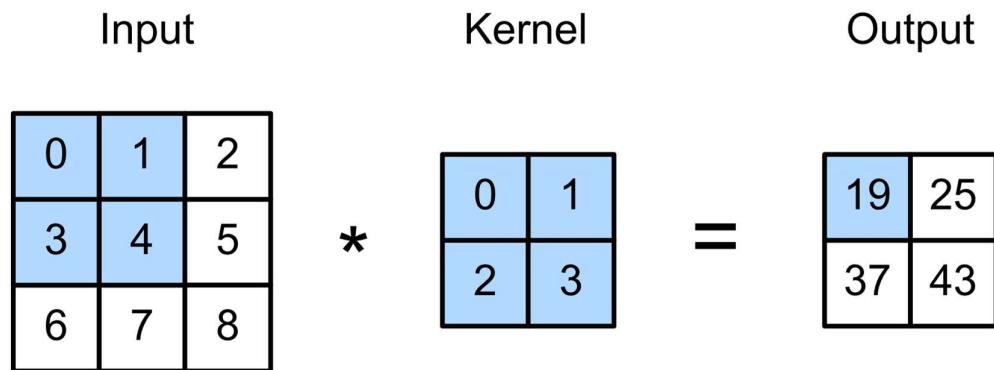
Mạng neuron tích chập sử dụng các hạt nhân ma trận (kernel matrix) để lọc ra các thuộc tính (feature) của dữ liệu. Việc này giúp giảm số lượng tính toán so với mạng neuron nhân tạo, bởi đặc trưng dữ liệu của mạng neuron tích chập là số lượng tính toán lớn (ảnh, văn bản, âm thanh, ...). Hơn nữa, việc sử dụng nhân ma trận giúp lưu trữ những thông tin liên kết giữa các giá trị đầu vào với nhau.

1.2.2. Phép Toán Tích Chập [11]

Trong phép toán tích chập (Convolution), một ma trận đầu vào và một ma trận hạt nhân được kết hợp bằng phép tính tương quan chéo (cross correlation) để tạo ra mảng đầu ra. Bắt đầu từ vị trí góc trên bên trái ($x = 0$ và $y = 0$), ma trận con và ma trận hạt nhân sử dụng phép toán tích vô hướng, kết quả là một giá trị số vô hướng duy nhất được ghi vào vị trí tâm (center) của ma trận con. Ma trận hạt nhân được xem như một cửa sổ trượt (Sliding Window) trên ma trận đầu vào.

Với ma trận đầu vào có kích thước $H \times W$, ma trận hạt nhân có kích thước $h \times w$, ta được ma trận mới có kích thước

$$(H - h + 1) \times (W - w + 1).$$



Hình 1.4 Phép Toán Tích Chập.

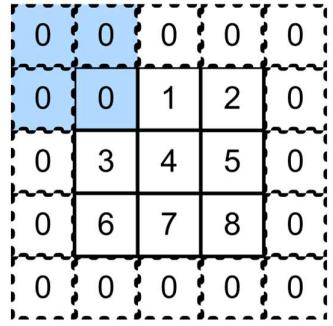
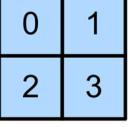
(Nguồn: <https://d2l.ai>)

1.2.3. Đếm Và Bước Nhảy [12]

Như đã nêu ở trên, kích thước ma trận đầu ra được xác định bởi kích thước của ma trận đầu vào mà ma trận hạt nhân. Theo công thức, ta có ma trận đầu ra sẽ có kích thước khác với ma trận đầu vào, cụ thể là ma trận đầu ra sẽ nhỏ hơn, dẫn tới mất một ít điểm ảnh trên biên của ma trận gốc. Việc mất mát này sẽ ảnh hưởng lớn nếu ta thực hiện phép tính tích chập nhiều lần và liên tiếp.

Để loại bỏ mất mát thông tin ở biên ma trận đầu vào, kỹ thuật Đếm (Padding) được thực hiện, bằng cách chèn thêm các giá trị xung quanh biên của ma trận đầu vào. Thông thường các giá trị thêm vào bằng 0. Như vậy kích thước đầu ra của ma trận đầu ra khi bước đếm có kích thước $p \times p$ là:

$$(H - h + 2p + 1) \times (W - w + 2p + 1)$$

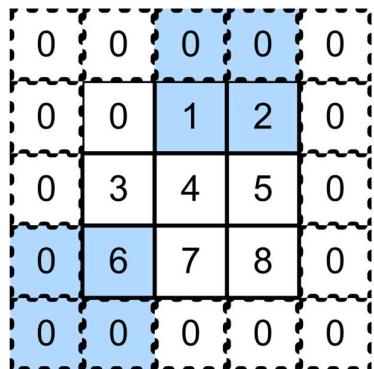
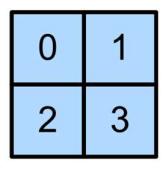
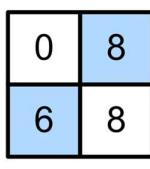
Input	Kernel	Output
		\ast = 

Hình 1.5 Phép Tích Tích Chập Với Bước Đệm = 1.

(Nguồn: <https://d2l.ai>)

Khi ma trận đầu vào có kích thước quá lớn, để tăng hiệu suất tính toán hoặc muốn giảm kích thước ma trận đầu ra, mỗi lần di chuyển ma trận hạt nhân, kỹ thuật Bước Nhảy (Stride) được thực hiện, bằng cách di chuyển ma trận hạt nhân nhiều hơn một, và bỏ qua những vị trí ở giữa. Khi bước nhảy s , ta có ma trận đầu ra kích thước là:

$$\left(\frac{H - h + 2p}{s} + 1\right) \times \left(\frac{W - p + 2p}{s} + 1\right)$$

Input	Kernel	Output
		\ast = 

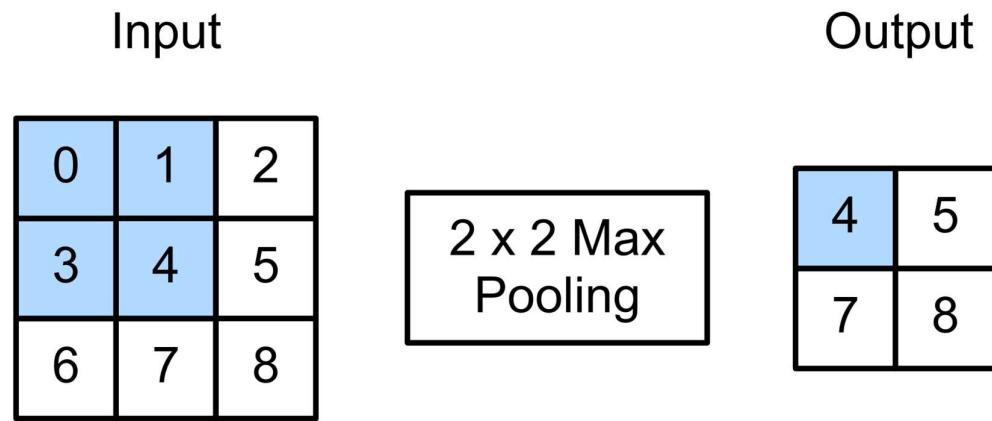
Hình 1.6 Phép Toán Tích Chập Với Bước Nhảy = 2.

(Nguồn: <https://d2l.ai>)

1.2.4. Hàm Gộp [13]

Hàm Gộp (Pooling Function) với mục đích giảm kích thước dữ liệu, giúp tối ưu tính toán trong quá trình huấn luyện, nhưng vẫn giữ được các thuộc tính quan trọng của ma trận đầu vào.

Giống như phép toán tích chập, phép toán gộp bao gồm một cửa sổ gộp (pooling window) trượt trên tất cả các vùng của ma trận đầu vào với giá trị bước nhảy (stride) nhất định, tính toán giá trị đầu ra duy nhất. Hai hàm gộp tiêu biểu là hàm gộp cực đại (max pooling) và hàm gộp trung bình (average pooling). Hàm gộp cực đại sử dụng giá trị lớn nhất trong ma trận trượt làm giá trị đầu ra. Trong khi hàm gộp trung bình sử dụng giá trị trung bình cộng các giá trị trong ma trận trượt. Như vậy giá trị đầu ra nhận được sự ảnh hưởng bởi các biến trong ma trận trượt, sẽ giữ được giá trị đặc trưng của ma trận đầu vào.

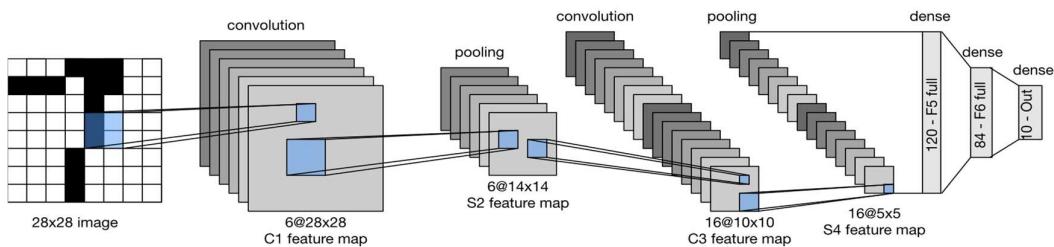


Hình 1.7 Phép Toán Gộp Cực Đại.

(Nguồn: <https://d2l.ai>)

1.2.5. Các Mô Hình Tích Chập Tiêu Biểu [14]

LeNet, mạng tích chập đầu tiên được giới thiệu năm 1989, phục vụ cho bài toán phân loại ảnh kí tự viết tay, trên tập dữ liệu MNIST.



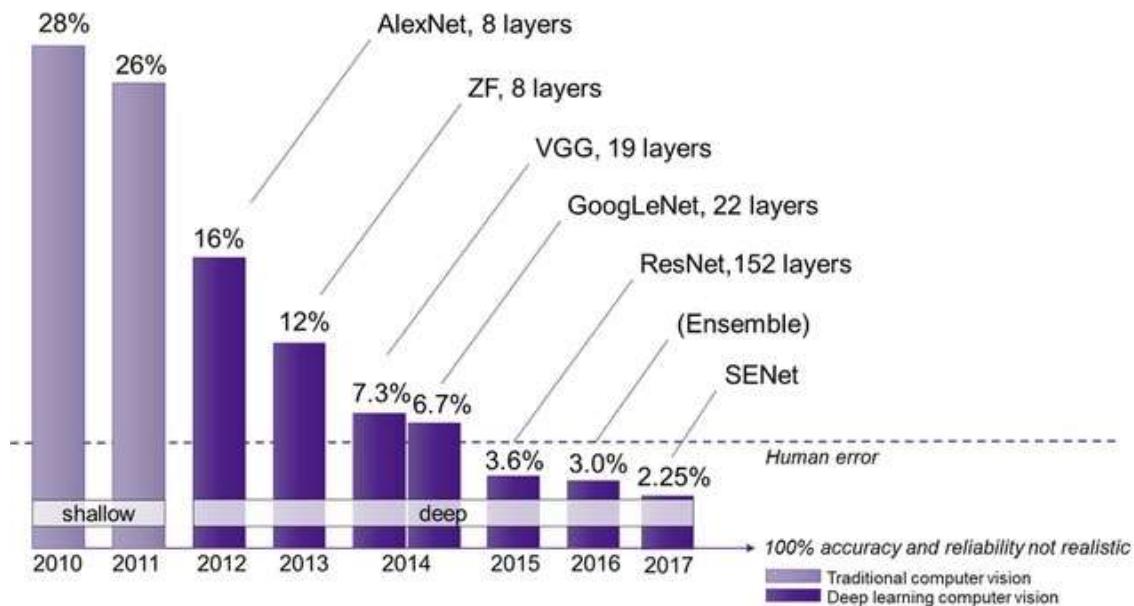
Hình 1.8 Kiến Trúc Mạng LeNet.

(Nguồn: <https://d2l.ai>)

Các phép toán tích chập và phép toán gộp đã được sử dụng. Tuy ở mức sơ khai, đây là nền tảng cho sự phát triển và cải tiến của các mạng neuron tích chập sau này.

Năm 2012 bắt đầu cho sự phát triển liên tục và vượt trội của mạng neuron tích chập. Lần lượt các mạng neuron tích chập AlexNet (2012), ZFNet (2013), VGG (2014), GoogleLeNet (2014) đạt được những cải tiến vượt bậc qua từng năm. Tuy nhiên độ chính xác vẫn chưa đạt được bằng con người.

Năm 2015, mạng ResNet đạt được độ chính xác tốt hơn con người (trên tập dữ liệu ảnh ImageNet). Những năm tiếp theo, những kiến trúc mạng vẫn tiếp tục tối ưu độ chính xác.



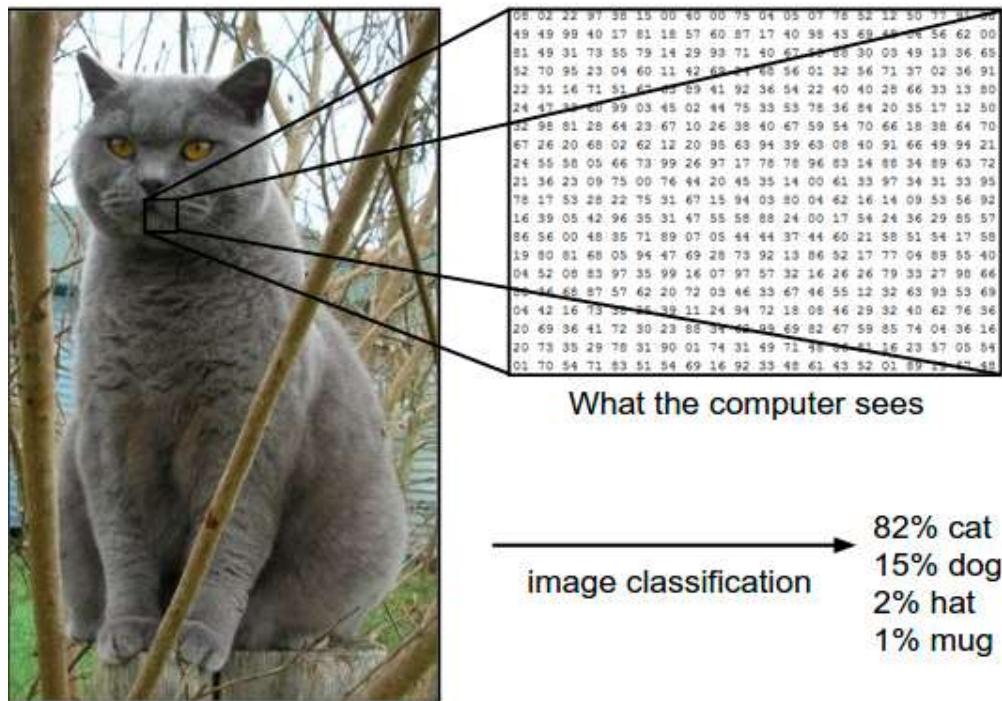
Hình 1.9 Sự Phát Triển Của Mạng Neuron Tích Chập.

(Nguồn: <https://semiengineering.com>)

1.3. Bài Toán Phân Loại Hình Ảnh

1.3.1. Sơ Lược Về Bài Toán Phân Loại Hình Ảnh

Bài toán phân loại hình ảnh (Image Classification) là một dạng bài toán quan trọng và phổ biến trong thị giác máy tính. Bằng việc sử dụng các mô hình mạng neuron tích chập, với dữ liệu đầu vào là ma trận nhiều chiều ứng với hình ảnh đầu vào và tập các nhãn, mô hình sẽ xác định nhãn của ma trận đầu vào đó.



Hình 1.10 Ví Dụ Về Bài Toán Phân Loại Hình Ảnh.

(Nguồn: <https://cuonglv1109.blogspot.com>)

1.3.2. Các Kiến Trúc Và Mô Hình Cho Bài Toán Phân Loại Hình Ảnh

Như đã trình bày ở mục 1.2, các kiến trúc cho bài toán phân loại hình ảnh được cải tiến và đạt được độ chính xác cao qua từng năm. Những mô hình tiêu biểu như AlexNet (2012), VGGNet (2014), ResNet (2016), SENet (2018), ...

1.3.3. Ứng Dụng Của Bài Toán Phân Loại Hình Ảnh

Kết quả bài toán phân loại hình ảnh có thể áp dụng vào rất nhiều lĩnh vực như phân loại động vật, phân loại biển báo giao thông hay nhận diện khuôn mặt, ... Phân loại ảnh cũng là bài toán cơ sở cho một số bài toán khác trong lĩnh vực Thị giác máy tính. Tập dữ liệu hình ảnh mà cộng đồng nghiên cứu bài toán này thường quan tâm là tập ImageNet. Tính đến tháng 2 năm 2018, tập dữ liệu này đã có hơn 14 triệu hình ảnh, thuộc hơn 20000 lớp khác nhau [15]. Điều này cho thấy sự phát triển và sức ảnh hưởng của bài toán phân loại hình ảnh.

1.3.4. Những Thách Thức Của Bài Toán Phân Loại Hình Ảnh

Những thách thức của bài toán phân loại hình ảnh như:

- Đa dạng về góc nhìn – Viewpoint variation: đối tượng cần phân loại khác nhau khi có góc nhìn khác nhau.

- Đa dạng về tỉ lệ và kích thước – Scale variation: Cùng một đối tượng cần phân loại tuy nhiên có thể có kích thước khác nhau.
- Biến dạng – Deformation: Sự đa dạng hình ảnh của cùng một đối tượng khi bị biến đổi theo các điều kiện khác nhau.

1.4. Bài Toán Nhận Diện Vật Thể

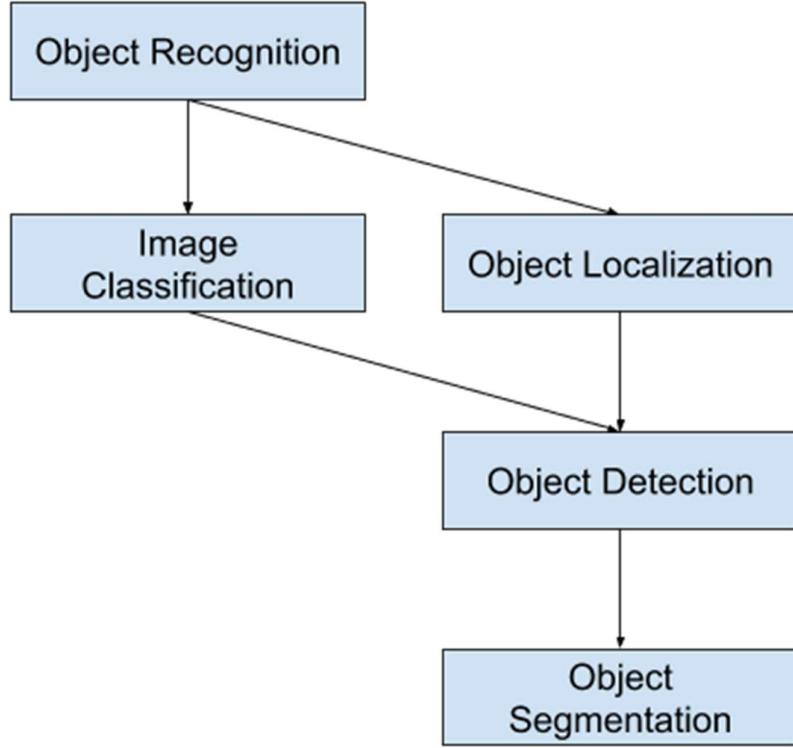
1.4.1. Sơ Lược Về Bài Toán Nhận Diện Vật Thể

Nhận diện vật thể (Object Detection) là một thuật ngữ chung để mô tả một tập hợp các nhiệm vụ thị giác máy tính có liên quan đến việc xác định các đối tượng trong ảnh kỹ thuật số. Để giải quyết được bài toán này ta cần giải quyết hai nhiệm vụ chính: phân loại hình ảnh (Image Classification) và định vị đối tượng (Object Localization). Bài toán phân loại hình ảnh đã được trình bày ở phần 1.3. Bài toán định vị đối tượng là bài toán xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box.

- Dữ liệu đầu vào: Một hình ảnh có một hoặc nhiều đối tượng.
- Dữ liệu đầu ra: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao.

Kết hợp giữa phân loại hình ảnh và định vị đối tượng, bài toán nhận diện vật thể là bài toán xác định vị trí hiện diện của các đối tượng trong bounding box và nhãn của các đối tượng nằm trong một hình ảnh.

- Dữ liệu đầu vào: Một hình ảnh có một hoặc nhiều đối tượng.
- Dữ liệu đầu ra: Một hoặc nhiều bounding box và nhãn cho mỗi bounding box.



Hình 1.11 Sơ Đồ Mối Liên Hệ Giữa Các Tác Vụ Trong Thị Giác Máy Tính.

(Nguồn: <https://phamdinhkhanh.github.io>)

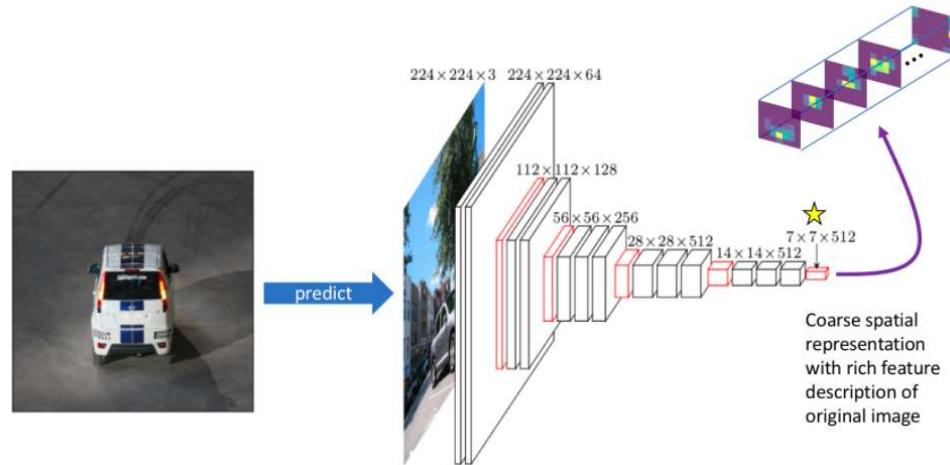
1.4.2. Các Loại Thuật Toán Và Các Mô Hình Tiêu Biểu

Bài toán nhận diện vật thể hiện tại được giải quyết theo hai phương pháp lớn: One-stage Object Detection và Two-stage Object Detection. Mỗi phương pháp có ưu điểm và nhược điểm riêng. Đối với One-stage có ưu điểm tốc độ xử lý nhanh, áp dụng cho các bài toán xử lý theo thời gian thực tốt, còn Two-stage có tốc độ xử lý chậm hơn nhưng lại chính xác hơn. Do vậy tùy thuộc vào ngữ cảnh bài toán mà ta chọn phương pháp giải quyết phù hợp.

Phương pháp One-stage [16]: ý tưởng của thuật toán là chia nhỏ hình ảnh thành các ô theo dạng lưới (grid) và đưa ra dự đoán vật thể từ các ô đó.

Để nhận biết thông tin của dữ liệu đầu vào, mô hình One-stage sử dụng một mạng neuron tích chập tiêu chuẩn (Standard Convolutional Network) để trích xuất những thông tin quan trọng của bức ảnh. Mạng neuron này còn được gọi là kiến trúc mạng “xương sống” (backbone), thông thường sẽ sử dụng những mạng neuron tích chập đã được huấn luyện trước để phân loại ảnh. Tuy nhiên những mạng tích chập này không có nhiệm vụ phân loại hình ảnh mà chỉ là trích xuất đặc trưng, vì vậy mạng neuron này

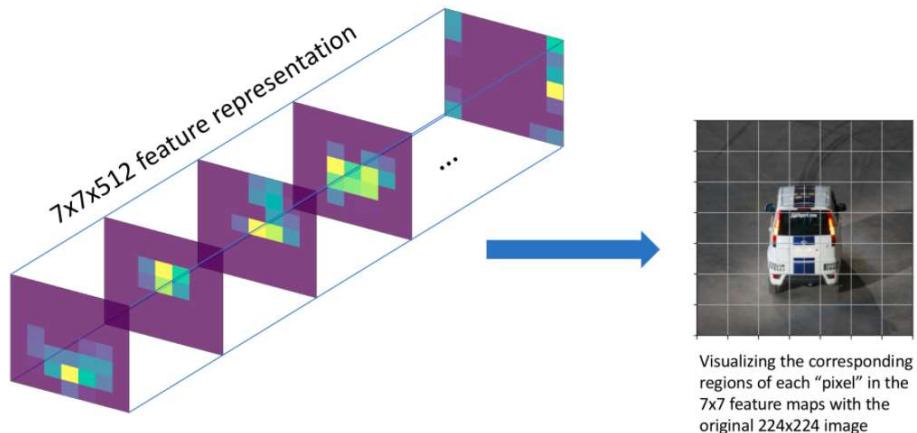
được loại bỏ một vài tầng cuối cùng, như vậy thông tin của hình ảnh được giữ lại ở độ phân giải thấp nhưng vẫn giữ được những thông tin đặc trưng.



Hình 1.12 Mô Hình Trích Xuất Đặc Trưng Trong Bài Toán Nhận Diện Vật Thể.

(Nguồn: <https://www.jeremyjordan.me>)

Với kiến trúc trên, ma trận đặc trưng từ những tầng cuối có thể biểu diễn dữ liệu đầu vào. Mỗi giá trị trong ma trận đầu ra đại diện cho một vùng giá trị trong ma trận đầu vào.



Hình 1.13 Mô Phỏng Giá Trị Đầu Ra Của Mạng Trích Xuất Đặc Trưng.

(Nguồn: <https://www.jeremyjordan.me>)

Từ ma trận đặc trưng này, mô hình có thể phát hiện và xác định gần đúng vị trí của đối tượng, và sử dụng những giá trị trong ô lưới đó cho nhiệm vụ phát hiện đối tượng.

Với ý tưởng trên, các mô hình One-stage đã phát triển với nhiều thuật toán đạt được độ chính xác cao và hiệu suất tốt như: Faster R-CNN, YOLO hay SSD.

Phương pháp Two-stage [17]: Kiến trúc của mạng Two-stage thường sử dụng hai thành phần mạng. Dữ liệu đầu vào sẽ đi qua mạng đề xuất (proposal network) để tìm kiếm và xác định đối tượng. Sau đó sử dụng mạng trích xuất đặc trưng, xử lý giá trị từ mạng đề xuất để tinh chỉnh đề xuất này và đưa ra dự đoán cuối cùng.

Phương pháp Two-stage đạt được độ chính xác cao nhưng nhược điểm của mô hình Two-stage sử dụng hai kiến trúc mạng, dẫn đến tốc độ thực thi chậm. Các cải tiến của mạng đề xuất được cải tiến khả năng xác định vật thể để có thể tối ưu thời gian tương tự như các mô hình của phương pháp One-stage.

Các mô hình Two-stage đạt hiệu suất cao như:

- R-CNN: Region Based CNN Detector (2014).
- Fast R-CNN: Faster Version Of R-CNN (2015).
- FPN: Feature Pyramid Network (2017).

1.4.3. *Ứng Dụng Của Bài Toán Nhận Diện Vật Thể*

Ứng dụng của bài toán nhận diện vật thể được áp dụng trong đa dạng các lĩnh vực trong cuộc sống hiện tại. Trong nhiều tác vụ Thị giác máy tính, hình ảnh đầu vào không chỉ chứa một vật thể cần nhận diện, mà thông thường sẽ có các ngoại cảnh chung với vật thể cần nhận diện. Khi đó bài toán phân loại hình ảnh không thể áp dụng, và phương pháp được áp dụng là nhận diện vật thể.

Một số ứng dụng tiêu biểu của bài toán nhận diện vật thể đã được áp dụng thực tế vào cuộc sống như:

- Face Detection: Bài toán xác định vị trí khuôn mặt là một bước quan trọng trong bài toán nhận diện khuôn mặt. Xác định vị trí khuôn mặt đã được ứng dụng rộng rãi trong việc chụp ảnh chân dung trên các thiết bị di động, giúp xác định vị trí cần lấy nét, cải thiện chất lượng ảnh chụp.
- Vehicle Detection: Bài toán nhận diện phương tiện giao thông được áp dụng để trực quan và tối ưu quá trình giám sát phương tiện giao thông. Nhận diện phương tiện giao thông cũng là bước đầu trong tác vụ đọc biển số phương tiện giao thông đang di chuyển, hay ứng dụng vào xe tự hành.
- Text Detection: Bài toán nhận diện chữ viết đóng vai trò quan trọng trong các tác vụ trích xuất thông tin chữ viết.

1.4.4. Những Thách Thức Của Bài Toán Nhận Diện Vật Thể

Ngoài những thách thức của bài toán phân loại hình ảnh đã được trình bày ở mục 1.3.4, bài toán nhận diện vật thể còn gặp phải những thách thức khác như:

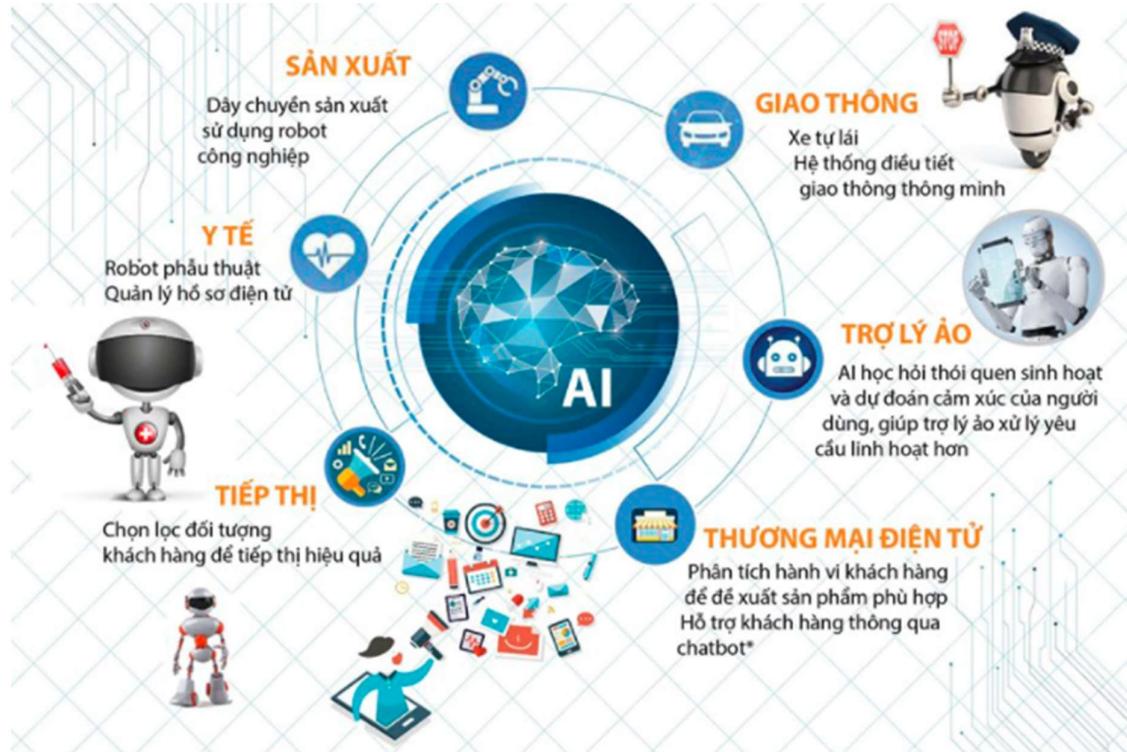
- Ánh hưởng bởi điều kiện ánh sáng – Illumination Conditions: khác với bài toán phân loại hình ảnh, bài toán nhận diện vật thể có nền ảnh và các đối tượng khác tác động, và điều kiện ánh sáng ảnh hưởng đến việc xác định vị trí chính xác của vật thể.
- Tác động của nền ảnh – Background Impaction: yếu tố nền ảnh ảnh hưởng đến việc xác định vị trí chính xác của vật thể. Khi nền ảnh phức tạp, việc xác định trở nên khó khăn và yêu cầu xử lý nhiều hơn.
- Độ phức tạp của mô hình – Complexity: việc xử lý hai tác vụ dẫn đến độ phức tạp của mô hình tăng, yêu cầu khối lượng xử lý nhiều hơn.
- Cân bằng giữa tốc độ và độ chính xác – Balance of Speed and Accuracy: Việc phát triển mô hình cân bằng giữa cả tốc độ thực thi, để đáp ứng xử lý thời gian thực, và độ chính xác của mô hình, vẫn chưa đạt được tốt nhất ở các mô hình hiện tại.
- Giới hạn dữ liệu – Limited Data: nguồn dữ liệu trong tác vụ nhận diện vật thể không nhiều, khó thu thập hơn. Hơn nữa, việc đánh nhãn dữ liệu yêu cầu độ chính xác cao, tốn nhiều thời gian thực hiện.

1.5. Những Ứng Dụng Khác Của Trí Tuệ Nhân Tạo Và Xử Lý Ảnh Số

Các giải pháp trong lĩnh vực xử lý ảnh số và trí tuệ nhân tạo có thể áp dụng cho đa dạng các bài toán trong thực tế ở các lĩnh vực khác nhau. Hiện nay, nhiều lĩnh vực đã đạt được những thành công như:

- Lĩnh vực y tế: xử lý ảnh y sinh, các ảnh chụp X-Quang, chẩn đoán bệnh, robot tự động phẫu thuật, gây mê, ...
- Lĩnh vực kinh tế: dự đoán thị trường, xu hướng kinh tế, nhu cầu tiêu dùng, ...
- Lĩnh vực giáo dục: xây dựng các phần mềm điểm danh, phát hiện gian lận thi cử, ...
- Lĩnh vực công cộng: phát hiện đám đông ở nơi công cộng, ...
- Lĩnh vực giao thông: nhận diện phương tiện, đo tốc độ phương tiện giao thông, ứng dụng xe tự lái, ...

- Lĩnh vực ngân hàng, tài chính: nhận diện và xác thực thông tin cá nhân, định danh khách hàng trực tuyến, ...
- Lĩnh vực dịch vụ: các ứng dụng tự động trả lời tin nhắn, tự động hoàn thành hồ sơ cá nhân ở khách sạn, sân bay, ...



Hình 1.14 Một Số Ứng Dụng Của Trí Tuệ Nhân Tạo

(Nguồn: <https://www.dienmayxanh.com>)

2. CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG

2.1. Dữ Liệu Hình Ảnh Giao Thông Việt Nam

Tập dữ liệu được thu thập từ Cổng Thông Tin Giao Thông Thành Phố Hồ Chí Minh [2]. Cụ thể như sau:

2.1.1. Về Thời Gian Thu Thập Dữ Liệu

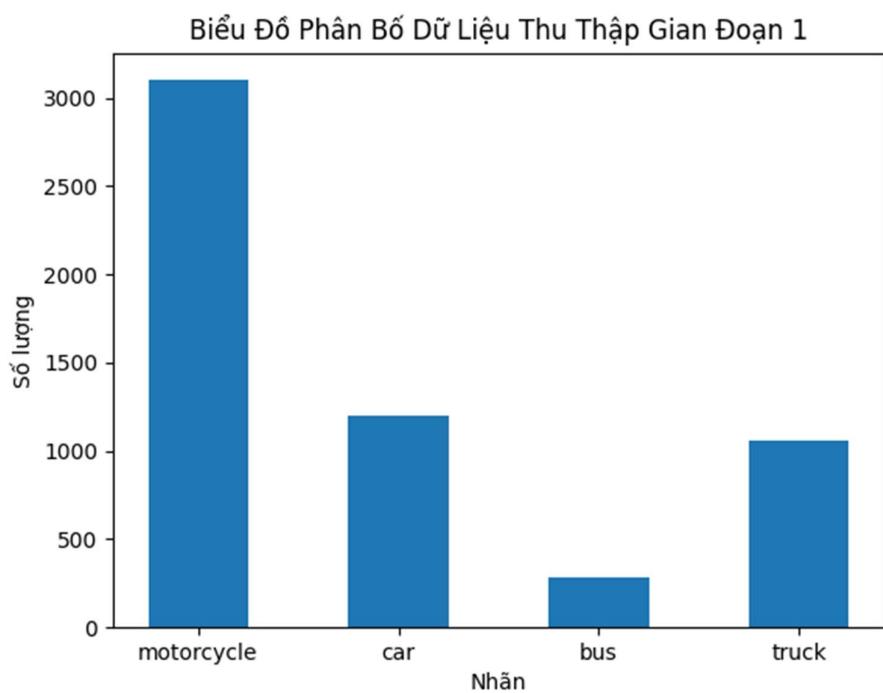
Dữ liệu được thu thập từ ngày 14/09/2021 đến ngày 30/11/2021. Dữ liệu được thu thập trong hai giai đoạn:

- Giai đoạn 1, từ ngày 14/09/2021 đến ngày 21/09/2021: thời gian Thành phố Hồ Chí Minh đang trong tình trạng giãn cách xã hội, số lượng xe lưu thông trên đường ít, số lượng xe tải lưu thông trong thành phố tăng. Tổng số lượng hình ảnh: 1580, tổng số lượng đối tượng: 5607, trung bình 3.54 đối tượng/ảnh.

Phân bố dữ liệu các lớp như sau:

Bảng 2.1 Bảng Phân Bố Dữ Liệu Thu Thập Giai Đoạn 1

Lớp	motorcycle	car	bus	truck
Số Lượng	3057	1206	287	1057

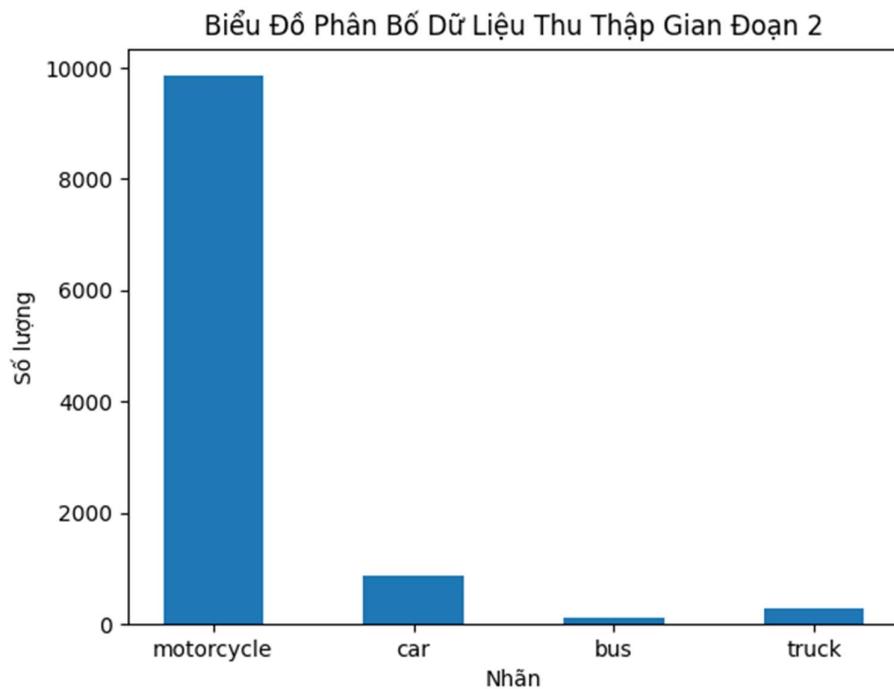


Hình 2.1 Biểu Đồ Phân Bố Dữ Liệu Thu Thập Giai Đoạn 1

- Giai đoạn 2, ngày 30/11/2021: thời gian thu thập từ 6 giờ đến 7 giờ 30 phút sáng. Số lượng xe máy chiếm phần lớn, lưu lượng giao thông đông. Tổng số ảnh: 1006, tổng số lượng đối tượng 11146, trung bình 11.08 đối tượng/ảnh. Phân bố dữ liệu các lớp như sau:

Bảng 2.2 Bảng Phân Bộ Dữ Liệu Thu Thập Giai Đoạn 2

Lớp	motorcycle	car	bus	truck
Số Lượng	9855	879	124	288



Hình 2.2 Biểu Đồ Phân Bộ Dữ Liệu Thu Thập Giai Đoạn 2

2.1.2. Về Phương Pháp Thu Thập

Trong giai đoạn 1, hình ảnh được thu thập bằng việc chụp hình ảnh hiển thị trên website của Cổng Thông tin Giao thông. Việc thu thập sử dụng thư viện PyAutoGUI, hỗ trợ chụp hình ảnh tự động, tuy nhiên hiệu suất không cao do không thể chụp từ nhiều camera. Hiệu suất đạt 13s/ảnh.

Trong giai đoạn 2, hình ảnh được thu thập từ API của Cổng Thông tin Giao Thông, sử dụng thư viện Tornado cho tác vụ lấy hình ảnh từ API và thư viện OpenCV cho tác vụ lưu hình ảnh. Việc thu thập này mang lại ưu điểm so với phương pháp cũ

như: có thể thu thập được nhiều camera cùng lúc. Hiệu suất cải thiện với việc thu thập đồng thời 6 camera, đạt 2.16s/ảnh.

2.1.3. Về Chất Lượng Hình Ảnh

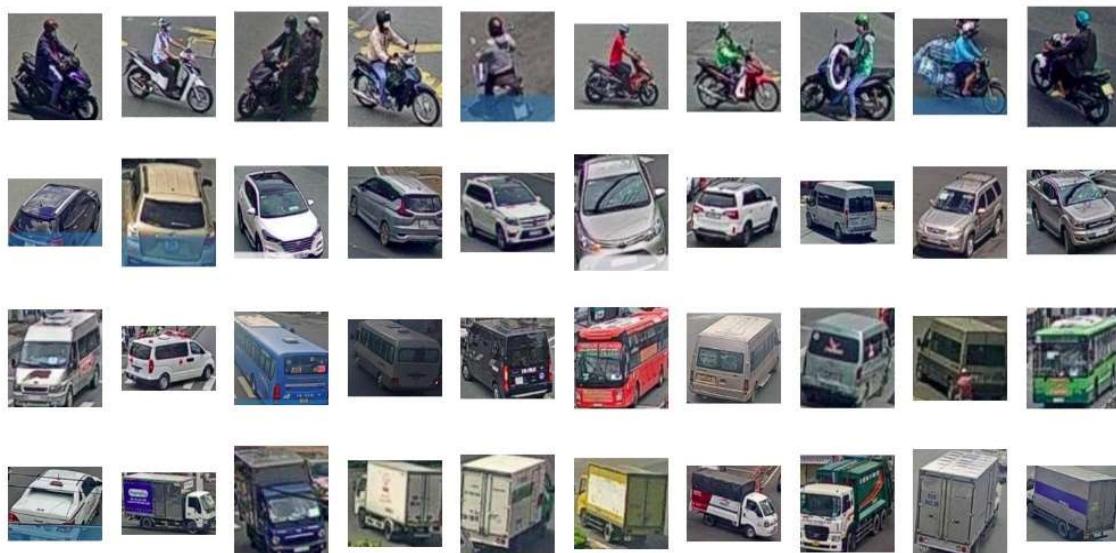
Chất lượng hình ảnh từ Cổng Thông tin Giao thông tùy thuộc vào từng camera. Có hai độ phân giải phổ biến là: 512×288 (pixel) và 800×450 (pixel). Dữ liệu hình ảnh được sử dụng trong bài báo cáo có độ phân giải 800×450 (pixel), được chọn từ 8 camera khác nhau.

2.1.4. Về Phân Loại Đối Tượng

Dữ liệu được chia thành 4 lớp:

- “motorcycle”: xe đạp, xe máy.
- “car”: xe hơi, xe bán tải
- “bus”: xe hơi lớn, xe buýt.
- “truck”: xe tải, xe container.

Hình ảnh đối tượng thuộc các lớp được mô tả dưới đây:

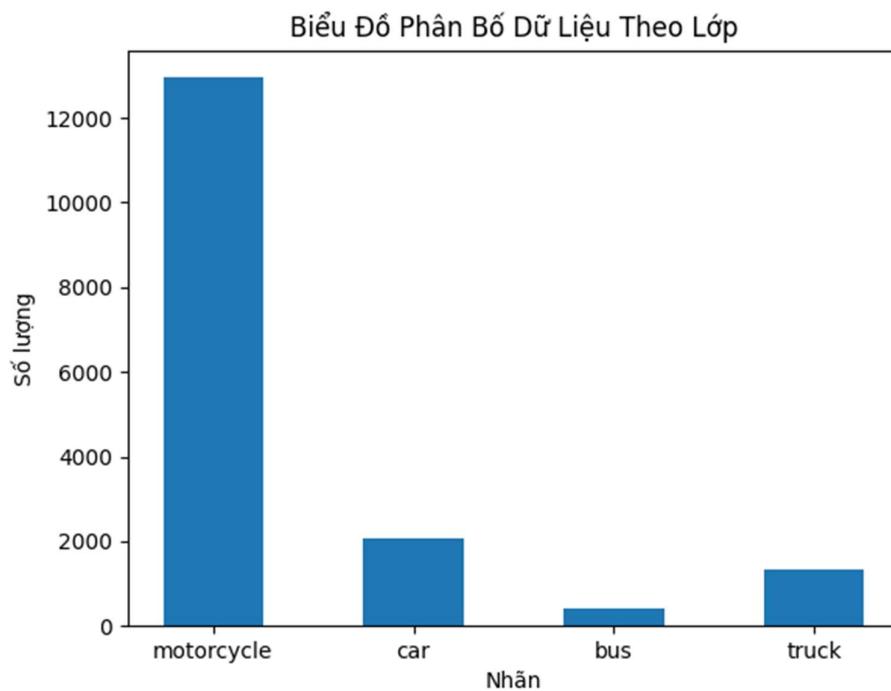


Hình 2.3 Hình Ảnh Các Phương Tiện Giao Thông Việt Nam

Phân bố dữ liệu theo lớp như sau:

Bảng 2.3 Bảng Phân Bố Dữ Liệu Theo Lớp

Lớp	motorcycle	car	bus	truck
Số Lượng	12912	2085	411	1345



Hình 2.4 Biểu Đồ Phân Bố Dữ Liệu Theo Lớp

2.1.5. *Đánh Nhãn Dữ Liệu*

Việc đánh nhãn dữ liệu thực hiện trên công cụ LabelImg. Việc thực hiện đánh nhãn do nhóm thực hiện.

Định dạng dữ liệu chuẩn Pascal VOC (Visual Object Classes). Pascal VOC là tập dữ liệu phục vụ cho bài toán nhận diện vật thể và các tác vụ khác trong xử lý ảnh. Chuẩn Pascal VOC lưu trữ thông tin nhãn dưới dạng tệp tin .xml (eXtensible Markup Language). Các thông tin quan trọng trong một tệp tin nhãn là:

- folder: lưu trữ thông tin về thư mục chứa ảnh của tập dữ liệu.
- path: đường dẫn đến ảnh tương ứng với tệp tin nhãn hiện tại.
- size: chứa 3 thông tin của ảnh là chiều rộng, chiều cao và chiều sâu.
- object: đánh dấu một đối tượng trong ảnh.
- name: tên lớp của đối tượng.
- bndbox: thông tin tọa độ của đối tượng đó, được lưu thành bộ bốn giá trị: x_min, y_min, x_max, y_max.

2.2. Chuẩn Hóa Và Tăng Cường Dữ Liệu

2.2.1. Chuẩn Hóa Dữ Liệu Hình Ảnh

Vì mô hình được lựa chọn là SSD 300 nên ảnh đầu vào cần chỉnh về kích thước 300×300 pixel. Hình ảnh được chuẩn hóa bằng cách tính hiệu giữa những điểm ảnh với giá trị trung bình màu sắc của tất cả điểm ảnh (Subtract Means).

Hình ảnh được tăng cường bằng cách sử dụng đồng thời những phương pháp như lật (Random Mirror), thay đổi giá trị và không gian màu (Photometric Distortion), cắt ảnh (Random Crop), mở rộng vùng đệm (Random Expand).

Ở các tập dữ liệu khác trên thế giới dành cho bài toán nhận diện vật thể, chiếm đa số là những hình ảnh với đối tượng có kích thước độ phân giải lớn, tỉ lệ chồng lấp không cao. Tuy nhiên, với tình trạng giao thông ở Việt Nam nói chung cũng như ở Thành Phố Hồ Chí Minh nói riêng, nhìn chung là mật độ giao thông cao. Bên cạnh đó, việc dữ liệu được lấy từ hệ thống camera giao thông, nên hình ảnh nhận về có chất lượng không cao, ảnh hưởng bởi chất lượng camera cũng như đường truyền tín hiệu, cộng với đặc thù là góc đặt cao nhưng hẹp về góc nhìn (bị ảnh hưởng bởi nhà cao tầng hoặc hệ thống đèn tín hiệu giao thông, hệ thống lưới điện).

Với những nhược điểm trên, các đối tượng nhỏ như xe máy sẽ có độ phân giải thấp, tỉ lệ chồng lấp cao và bị khuất một phần bởi các phương tiện có kích thước lớn hơn. Việc áp dụng các yếu tố tăng cường dữ liệu không thực sự hiệu quả trên đối tượng xe máy.



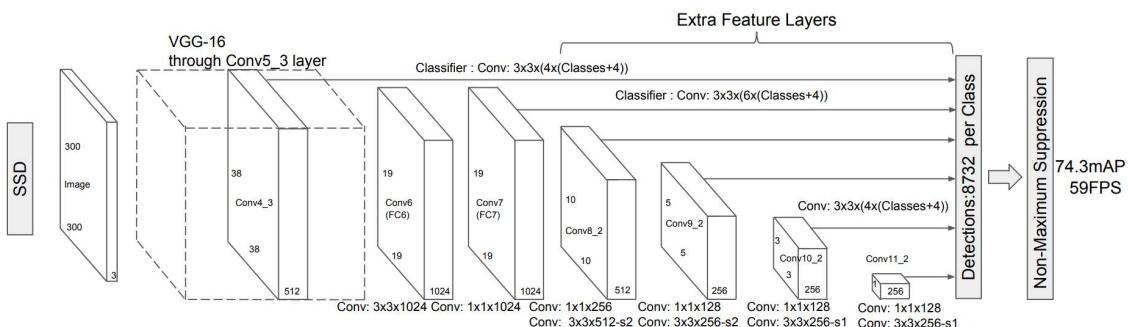
Hình 2.5 Các Phép Toán Chuẩn Hóa Và Tăng Cường Dữ Liệu

3. CHƯƠNG 3. NHẬN DIỆN VẬT THỂ VỚI MÔ HÌNH SSD

3.1. Tổng Quan Về Mô Hình SSD

Single Shot MuiliBox Detector (SSD) [18] được công bố cuối năm 2016, một mô hình cho bài toán nhận diện vật thể. SSD được thiết kế và phát triển cho các tác vụ cần thời gian thực thi nhanh, phát hiện đối tượng theo thời gian thực (real-time).

SSD được thiết kế sử dụng một mạng neuron đơn (single deep neural network), loại bỏ mạng đề xuất (proposal network) và co giãn kích thước trạng thái (feature resampling stages), thay thế bằng cách sử dụng các lớp tích chập với kích thước nhỏ để phân loại vật thể và dự đoán vị trí và đóng gói tất cả tính toán vào trong một kiến trúc mạng. Những cải tiến trên giúp việc huấn luyện mô hình SSD dễ dàng hơn, tích hợp vào các hệ thống đơn giản và giảm đáng kể thời gian dự đoán.



Hình 3.1 Kiến Trúc Mô Hình SSD [18]

Bằng việc sử dụng nhiều tầng trích xuất đặc trưng với các kích thước khác nhau, SSD đạt được độ chính xác cao khi sử dụng dữ liệu đầu vào ở kích thước nhỏ. Tốc độ thực thi cũng giảm đáng kể từ những cải tiến trên. Cụ thể, trên tập dữ liệu VOC2007, mô hình SSD đạt được 59 FPS với độ chính xác mAP 74.3%, so sánh với Faster R-CNN là 7 FPS với độ chính xác mAP 73.2%, và YOLO là 45 FPS và 63.4% mAP, các số liệu so sánh sẽ được trình bày chi tiết ở mục 3.6.

SSD sử dụng một mạng neuron tích chập tiêu chuẩn cho việc trích xuất đặc trưng. Mạng neuron này sử dụng các kiến trúc tiêu chuẩn sử dụng cho bài toán phân loại hình ảnh, và loại bỏ các tầng có nhiệm vụ phân loại hình ảnh (classification layers), sau đó thêm các module phụ trợ cho việc dự đoán vị trí vật thể và phân loại vật thể. Trong bài báo của nhóm tác giả, mô hình mạng neuron sử dụng cho tầng trích xuất đặc trưng là VGG-16. Các mô hình khác vẫn cho kết quả tốt tương tự.

3.2. Kiến Trúc Mạng VGG

Kiến trúc mạng VGG [19] được giới thiệu năm 2014, do nhóm tác giả Visual Geometry Group, thuộc đại học Oxford giới thiệu. Tên gọi VGG được đặt theo tên nhóm tác giả.

VGG là một kiến trúc mạng neuron tích chập. Với mô hình VGG, quan điểm về một mạng neuron sâu sẽ cải thiện độ chính xác của mô hình. Thực tế, mô hình VGG-16 đạt 91.9% top-5 test trên tập dữ liệu hình ảnh ImageNet [20], gồm 14 triệu hình ảnh thuộc 1000 lớp khác nhau.

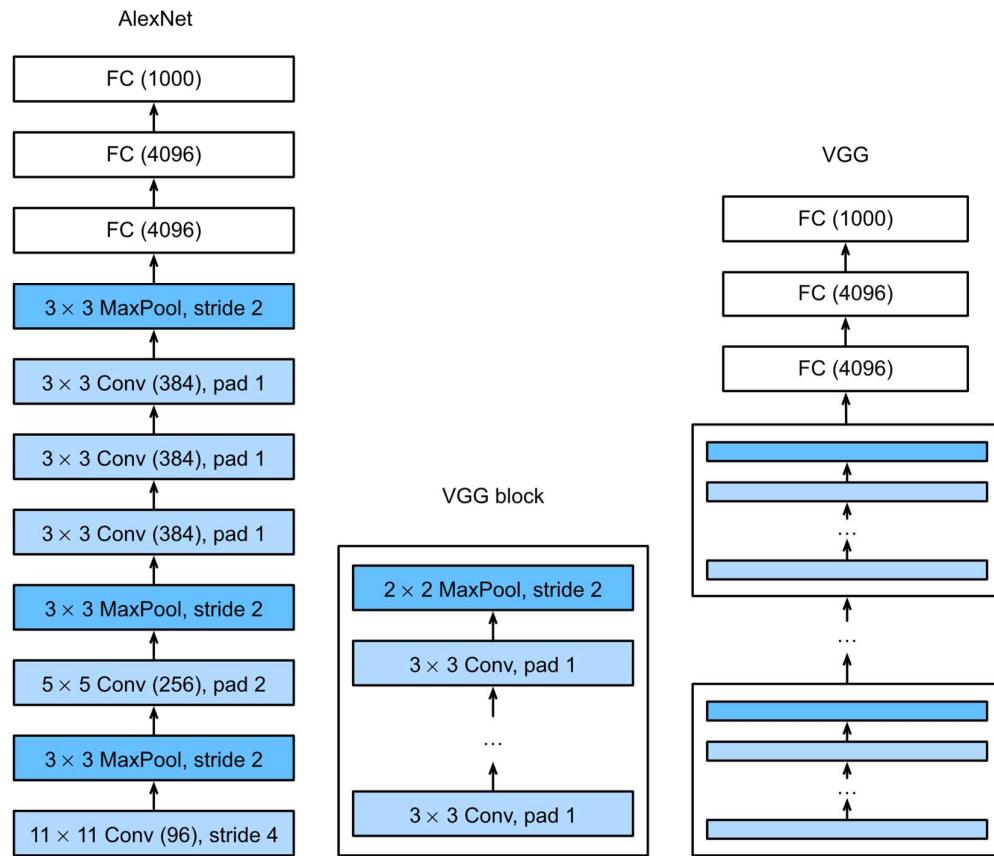
3.2.1. *Khối VGG*

Một khối cơ bản của mạng neuron tích chập gồm:

- Phép toán tích chập với phân đệm nhằm duy trì kích thước ma trận.
- Hàm kích hoạt như ReLU hay Sigmoid.
- Hàm gộp như phép gộp cực đại hay phép gộp trung bình, nhằm giảm chiều dữ liệu.

Kiến trúc này được áp dụng trong các mô hình trước VGG như LeNet hay AlexNet. Nắm mục tiêu trích xuất nhiều đặc trưng hơn, một khối VGG sử dụng nhiều hơn một phép toán tích chập và hàm kích hoạt, kết thúc bởi một tầng gộp cực đại. Cụ thể, nhóm tác giả sử dụng:

- Phép toán tích chập với ma trận hạt nhân 3×3 , bước đệm 1.
- Hàm kích hoạt phi tuyến ReLU.
- Hàm gộp cực đại với ma trận hạt nhân 2×2 , bước nhảy 2.

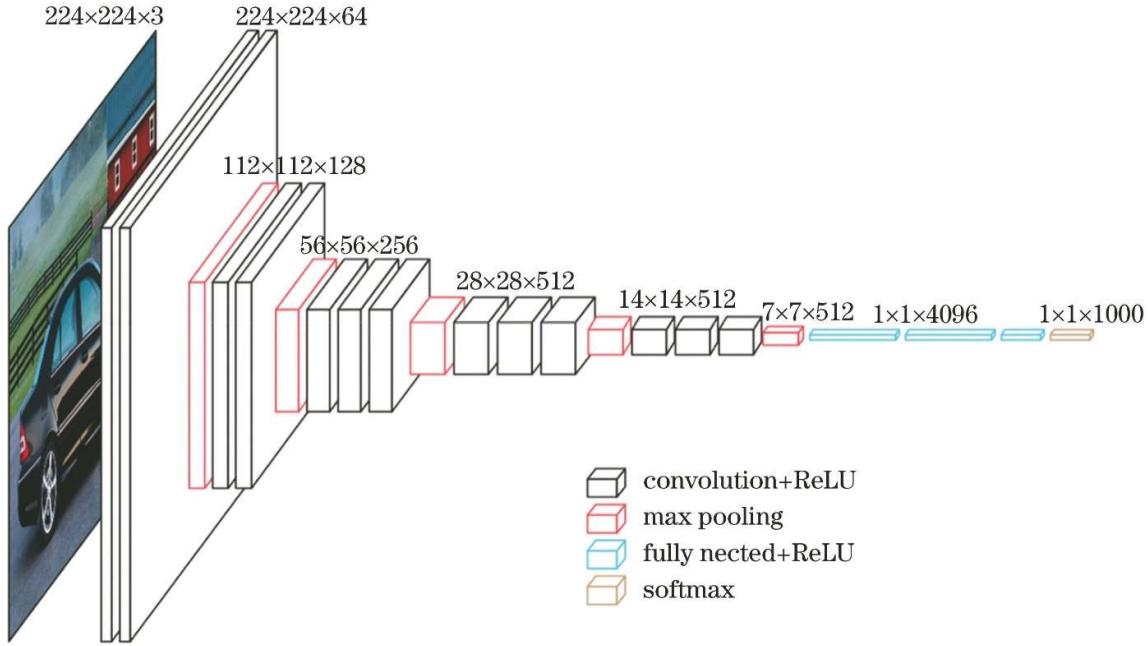


Hình 3.2 So Sánh Mạng VGG Và Mạng AlexNet.

(Nguồn: <https://d2l.ai>)

3.2.2. Mạng VGG

Mạng VGG được kết hợp bởi hai thành phần: thành phần đầu tiên bao gồm các lớp tích chập, thành phần thứ hai gồm các tầng kết nối đầy đủ. Kiến trúc mạng VGG-16 gồm năm khối tích chập, hai khối đầu tiên gồm hai phép toán tích chập, ba tầng tiếp theo gồm ba phép toán tích chập. Cuối cùng được kết nối với ba tầng kết nối đầy đủ. Con số 16 biểu hiện số tầng tích chập và số tầng kết nối đầy đủ trong mạng, ngoài ra còn có các biến thể khác như VGG-11, VGG-13 hay VGG-19.



Hình 3.3 Kiến Trúc Mô Hình Mạng VGG-16.

(Nguồn: <https://www.researchgate.net>)

3.2.3. *Ưu điểm, nhược điểm*

Kiến trúc mạng VGG-16 sâu hơn so với mạng AlexNet (13 tầng tích chập thay vì 5 tầng tích chập), và có số lượng tham số lớn hơn, lên đến 138 triệu tham số. Đây là một trong những mạng có số lượng tham số lớn nhất. Với lượng thông tin học nhiều, VGG đạt được độ chính xác cao. Với kiến trúc đơn giản, tốc độ tính toán của VGG khá nhanh, dễ dàng triển khai và thay đổi cấu trúc mạng.

Bắt đầu từ kiến trúc VGG với khối VGG, đã hình thành một hình mẫu chung cho các mạng tích chập sau này, đó là mạng sẽ trở nên sâu hơn, lượng thông tin học nhiều hơn và sử dụng các khối có kiến trúc tương tự khối VGG.

Nhược điểm của mạng VGG do kiến trúc sâu và nhiều tham số gây ra. Với số lượng tham số lớn, mô hình sẽ tốn chi phí cao hơn để huấn luyện, nếu không sử dụng các mô hình đã huấn luyện trước đó (pretrained model). Hơn nữa, do kiến trúc sâu, hiện tượng triệt tiêu đạo hàm (Vanishing Gradient) xảy ra. Các kiến trúc mạng về sau với các kiến trúc đa nhánh đã khắc phục được vấn đề trên.

3.2.4. *Mô Hình VGG Trong SSD*

Về cơ bản, mô hình SSD là sự kết hợp của hai thành phần, là thành phần trích xuất đặc trưng hình ảnh, và thành phần nhận diện hình ảnh. Trong bài báo được công bố, thành phần trích xuất đặc trưng được sử dụng là mô hình VGG-16.

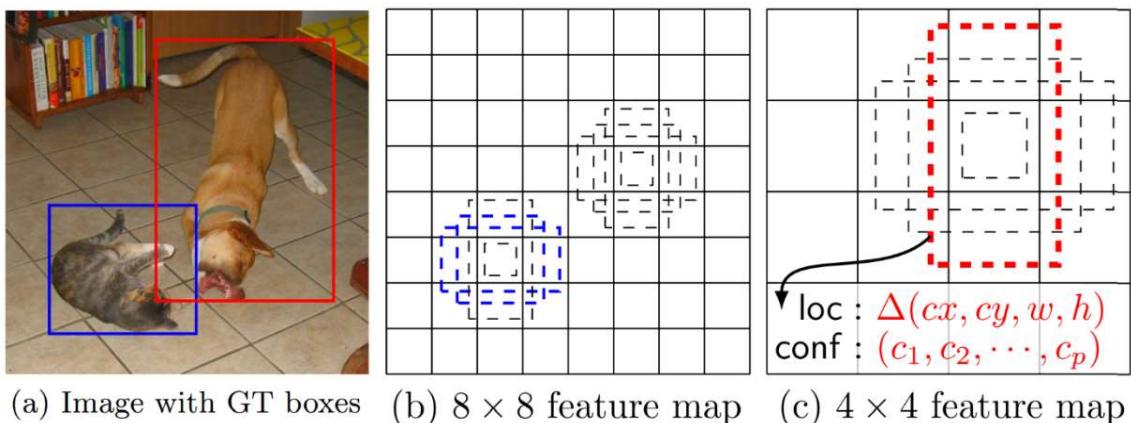
Với kiến trúc SSD có dữ liệu đầu vào kích thước 300×300 , mô hình trích xuất đặc trưng VGG thay đổi thông số giá trị đầu vào từ 224×224 tăng lên 300×300 , loại bỏ các tầng phân loại ở cuối mạng gốc, tăng thêm hai lớp tích chập, cũng như lấy giá trị từ khối tích chập thứ tư và ở cuối của mạng, nhằm phục vụ cho quá trình nhận diện vật thể ở đa dạng kích thước.

3.3. Các Module Phụ Trợ

3.3.1. Module Trích Xuất Đặc Trưng

Sau khi qua mô hình mạng trích xuất đặc trưng, cụ thể, khi dữ liệu đầu vào đi qua mô hình mạng VGG-16, ta sẽ thu được hai ma trận đặc trưng ở giữa và cuối mạng. Hai ma trận đặc trưng này theo cách triển khai của nhóm tác giả có kích thước lần lượt là 38×38 và 19×19 . Ở cuối mạng VGG, sau khi đã loại bỏ các thành phần phân loại, kết quả ma trận đầu ra tiếp tục đi qua những lớp tích chập. Nhiệm vụ của các lớp tích chập này trích xuất được những đặc trưng ở những kích thước nhỏ hơn. Những lớp này có xu hướng giảm dần kích thước, nhờ đó mô hình có thể đưa ra kết quả dự đoán ở nhiều tỉ lệ khác nhau.

Từ giá trị đầu ra của mạng VGG, module trích xuất đặc trưng sẽ tạo ra bốn ma trận đặc trưng có kích thước lần lượt là 10×10 , 5×5 , 3×3 và 1×1 . Bằng việc sử dụng các lớp tích chập với ma trận hạt nhân kích thước nhỏ (1×1 và 3×3), cùng với việc không sử dụng hàm gộp, module trích xuất đặc trưng hoạt động hiệu quả với các ma trận đặc trưng có kích thước nhỏ. Mô hình SSD sử dụng các ma trận đặc trưng nhỏ để dự đoán những vật thể có kích thước lớn, và ngược lại với ma trận đặc trưng lớn và vật thể có kích thước nhỏ.



Hình 3.4 Mô Phỏng Ma Trận Đặc Trưng Với Tỉ Lệ Khác Nhau [18]

Hình 3.4 mô phỏng sự hiệu quả khi mô hình SSD sử dụng nhiều kích thước khác nhau cho mô hình trích xuất đặc trưng. Đối tượng Chó (màu đỏ) phù hợp với một tọa độ nào đó (ô nét đứt màu đỏ) trong ma trận đặc trưng kích thước 4×4 , nhưng lại không phù hợp với bất kỳ vị trí nào trong ma trận có kích thước to hơn là 8×8 , ngược lại với đối tượng Mèo (màu xanh) trong ma trận đặc trưng 8×8 .

Kết quả thực nghiệm cho thấy, việc sử dụng nhiều ma trận đặc trưng với các tỉ lệ khác nhau cải thiện độ chính xác. Kết quả độ chính xác trên thang đo mAP tăng từ 62.4% lên 74.6%.

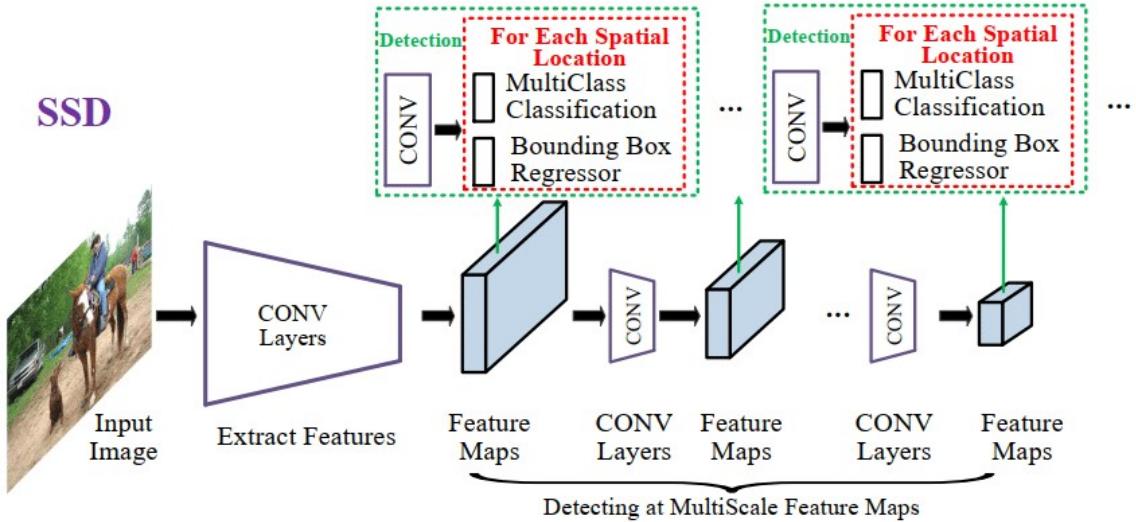
3.3.2. *Module Phân Loại Vật Thể*

Module phân loại vật thể sử dụng các kết quả từ mô hình trích xuất đặc trưng. Các ma trận đặc trưng tạo ra một bộ các giá trị dự đoán bằng cách sử dụng một bộ các ma trận tích chập hạt nhân trượt trên ma trận đặc trưng. Với ma trận đặc trưng có kích thước $m \times n$ và p kênh, một ma trận tích chập hạt nhân kích thước $3 \times 3 \times p$ trượt trên ma trận đặc trưng, tạo ra một bộ giá trị gồm tỷ lệ dự đoán của vật thể thuộc một lớp, cũng như thông tin về các kích thước vị trí liên quan đến một vị trí mặc định nào đó. Như vậy, tại mỗi vị trí trong bộ $m \times n$ của ma trận đặc trưng, tạo ra một bộ giá trị gồm thông tin đến tỷ lệ dự đoán vật thể thuộc vào c lớp, và bốn thông tin vị trí liên quan đến xác định vị trí. Giá trị trả về bao gồm ma trận có kích thước $m \times n \times c$ cho tác vụ phân loại vật thể và ma trận có kích thước $m \times n \times 4$ cho tác vụ xác định vị trí vật thể.

3.3.3. *Module Dự Đoán Vị Trí Vật Thể*

Với mỗi ma trận đặc trưng, mô hình SSD sử dụng một bộ vị trí mặc định (default box). Vị trí mặc định này sẽ giúp tính toán vị trí chính xác cụ thể của vật thể cần tìm. Bốn thông tin mà module dự đoán vị trí trả về bao gồm độ chênh lệch hoành độ (Δcx), độ chênh lệch tung độ (Δcy), kích thước chênh lệch theo chiều dài (Δw) và kích thước chênh lệch theo chiều rộng (Δh). Các thông tin này được tính dựa trên vị trí tâm và kích thước của vị trí mặc định.

Với ma trận đặc trưng kích thước $m \times n$, số lượng vị trí mặc định k và số lượng lớp cần phân loại c , ta cần tính toán tổng cộng $(c + 4)kmn$ phép tính toán.



Hình 3.5 Module Xác Định Vị Trí Và Phân Loại Vật Thể

(Nguồn: <https://www.researchgate.net>)

3.4. Thuật Toán Sinh Default Boxes

Việc xây dựng một bộ các ô mặc định (default box) với nhiều kích thước, tỉ lệ và từ các ma trận đặc trưng là phương pháp quan trọng và đạt độ hiệu quả cao trong mô hình SSD. Các default box này sẽ xử lý việc nhận dạng các vật thể với kích thước khác nhau tối ưu hơn các phương pháp trước đây như thay đổi kích thước dữ liệu đầu vào với các tỉ lệ phóng to hay thu nhỏ khác nhau.

Một số định nghĩa trong việc xây dựng bộ default box:

- Scale: Độ phóng đại của khung hình gốc. Ví dụ: Nếu khung hình gốc có giá trị là (w, h) thì sau phóng đại khung hình với tỉ lệ s , khung hình mới có kích thước là (sw, sh) , với miền giá trị $s \in (0,1]$. Scale sẽ kết hợp với aspect ratio để nhận được các khung hình có tỷ lệ cạnh w/h khác nhau.
- Aspect ratio: Tỷ lệ cạnh, được đo bằng tỷ lệ giữa chiều dài và chiều rộng của khung hình, nhằm xác định hình dạng tương đối của khung hình bao chúa vật thể. Chẳng hạn nếu vật thể là người thường có aspect ratio = 1:3 hoặc xe cộ nhìn từ phía trước là 1:1.
- Bounding box: Khung hình bao chúa vật thể được xác định trong quá trình huấn luyện.
- Ground truth box: Khung hình bao chúa vật thể đã được xác định (gắn nhãn).

- Offsets: bộ giá trị $(\Delta cx, \Delta cy, \Delta w, \Delta h)$, dùng để xác định vị trí của vật thể từ giá trị của default box.
- IoU: tỉ lệ Intersection of Union là tỉ lệ đo lường mức độ giao nhau giữa 2 khung hình (thường là khung hình dự báo và khung hình thực tế) để nhằm xác định 2 khung hình trùng lặp không. Tỉ lệ này được tính dựa trên phần diện tích giao nhau và không giao nhau giữa chúng.

Các default box được khởi tạo thông qua giá trị scale và aspect ratio tại mỗi tầng ma trận đặc trưng.

Với mỗi ma trận đặc trưng, mô hình SSD sẽ xác định một tỉ lệ scale tương ứng. Với ma trận đặc trưng thứ k trong tổng số m ma trận đặc trưng, tỉ lệ scale s_k được xác định theo công thức:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1), \quad k \in [1, m]$$

Giá trị $s_{min} = 0.2$ và $s_{max} = 0.9$, các giá trị này có thể thay đổi tùy theo bộ dữ liệu để đạt giá trị tốt nhất. Tại ma trận đặc trưng đầu tiên, giá trị $s_k = 0.2$, trong các ma trận đặc trưng tiếp theo, giá trị này tăng tuyến tính và tại ma trận đặc trưng cuối cùng, giá trị $s_k = 0.9$.

Kết hợp với một giá trị s_k xác định tại mỗi ma trận đặc trưng, hệ số aspect ratio được áp dụng để tạo ra những default box có các tỉ lệ khác nhau. Tập giá trị của hệ số aspect ratio $a_r = \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$. Tại ma trận đặc trưng thứ k , tỉ lệ scale s_k , ta có độ dài và độ rộng của default box đó được xác định bởi:

$$w_k^a = s_k \sqrt{a_r}, \quad h_k^a = s_k \sqrt{a_r}$$

Tại giá trị aspect ratio bằng 1, mô hình tạo thêm một default box với giá trị scale mới được tính bằng:

$$s'_k = \sqrt{s_k s_{k+1}}$$

Tổng cộng sẽ có 6 default box cho mỗi vị trí trong ma trận đặc trưng. Tuy nhiên, theo thiết kế, các ma trận đặc trưng thứ 1, 5 và 6 sẽ loại bỏ hai tỷ lệ aspect ratio 3 và $\frac{1}{3}$, vì thế số lượng default box tại mỗi vị trí ở các tầng này chỉ là 4 default box.

Như đã trình bày trong phần 3.3.2, một ma trận tích chập hạt nhân kích thước $3 \times 3 \times p$ trượt trên ma trận đặc trưng, số lượng default box được tạo ra tại mỗi ma trận đặc trưng được trình bày trong bảng 3.1.

Bảng 3.1 Bảng Giá Trị Số Lượng Default Box Mỗi Ma Trận Đặc Trung

	Ma Trận Đặc Trung	Kích Thước Đầu Vào	Số Lượng Tỷ Lệ Aspect Ratio	Số Lượng Default Box
1	Conv4_3	38×38	4	$38 \times 38 \times 4 = 5776$
2	Conv7	19×19	6	$19 \times 19 \times 6 = 2166$
3	Conv8_2	10×10	6	$10 \times 10 \times 6 = 600$
4	Conv9_2	5×5	6	$5 \times 5 \times 6 = 150$
5	Conv10_2	3×3	4	$3 \times 3 \times 4 = 36$
6	Conv11_2	1×1	4	$1 \times 1 \times 4 = 4$

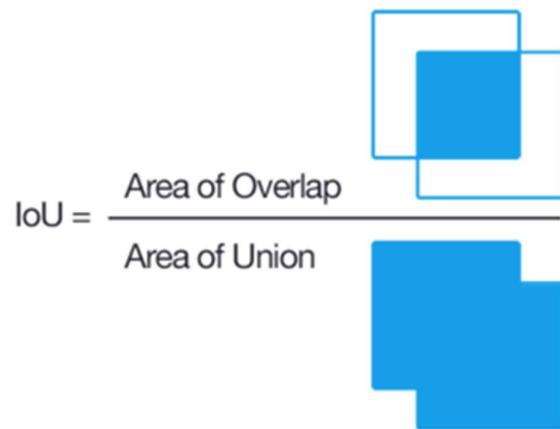
Mô hình SSD sẽ tạo ra 8732 default box với các tỉ lệ và kích thước khác nhau. So sánh giá trị này với mô hình YOLO là $7 \times 7 \times 2 = 98$, số lượng default box của SSD nhiều hơn đáng kể.

3.5. Huấn Luyện Mô Hình Nhận Diện Vật Thể

3.5.1. Chiến Thuật Huấn Luyện

Mô hình SSD được xây dựng như một mạng nơ ron tích chập đơn, vì vậy SSD chỉ cần dữ liệu đầu vào và nhãn của dữ liệu đó. Cụ thể hơn là hình ảnh đầu vào với kích thước xác định, và bộ các giá trị nhãn (ground truth box). Trong quá trình huấn luyện, mục tiêu của mô hình SSD là tìm ra default box nào phù hợp với ground truth box.

Với mỗi ground truth box, sẽ có nhiều default box phù hợp, với đa dạng các vị trí, tỉ lệ, độ co giãn khác nhau. Mô hình sẽ tính toán hệ số trùng khớp Jaccard [21] giữa ground truth box và từng default box. Những default box có giá trị lớn hơn ngưỡng (theo bài viết của nhóm tác giả cài đặt ngưỡng bằng 0.5) sẽ được sử dụng để dự đoán vật thể. So sánh với bài báo “Scalable Object Detection Using Deep Neural Networks” [22], khi mô hình cố gắng tìm ra một giá trị default box tốt nhất với độ trùng khớp cao nhất, mô hình SSD tính toán tất cả các default box tốt hơn ngưỡng, như vậy sẽ đơn giản và tối ưu hơn trong quá trình huấn luyện, cũng như cho phép mô hình dự đoán được độ chính xác cao trên nhiều default box, hơn là chọn một default box có độ trùng khớp cao nhất. Hệ số Jaccard trong bài toán nhận diện vật thể còn được gọi là IoU (Intersection of Union).



Hình 3.6 Minh Họa Hàm số IoU

(Nguồn: wikipedia.org)

Miền giá trị của hệ số Jaccard trong đoạn [0,1], với giá trị càng cao thể hiện giá trị dự đoán và giá trị nhãn trùng khớp cao.



Hình 3.7 Minh Họa Giá Trị IoU

(Nguồn: wikipedia.org)

3.5.2. Hàm Số Non-Maximum Suppression

Với số lượng lớn default box được tạo ra, cũng như giá trị của những bounding box gần nhau cùng dự đoán cho một vật thể, làm cho lượng thông tin tính toán quá lớn, không đem lại hiệu quả tính toán. Chúng ta chỉ cần một bounding box để thể hiện một đối tượng. Hàm số Non-Maximum Suppression (NMS) được xây dựng nhằm mục đích tìm được bounding box tốt nhất, và loại bỏ những bounding box có tỉ lệ trùng lặp cao.

Giá trị đầu vào của hàm số NMS gồm có:

- Mảng các giá trị bounding box B .

- Mảng các giá trị độ tự tin S tương ứng với các bounding box B .
- Giá trị ngưỡng trùng lắp N_t .

Giá trị đầu ra của hàm số NMS là mảng các giá trị bounding box D đã loại bỏ các giá trị bounding box có độ trùng lắp cao.

Thuật toán NMS:

1. Tìm giá trị bounding box có độ tự tin cao nhất, loại bỏ bounding box này khỏi mảng bounding box đầu vào B và thêm vào mảng bounding box kết quả D (Khởi tạo mảng D rỗng).
2. So sánh bounding box trên với tất cả các bounding box còn lại trong B , tính toán giá trị IoU giữa giá trị lớn nhất và từng giá trị còn lại trong B . Nếu giá trị IoU này lớn hơn ngưỡng trùng lắp N , loại bỏ giá trị này trong B .
3. Lặp lại bước 1 và bước 2 với mảng B đã loại bỏ các giá trị có độ trùng lắp lớn hơn N .
4. Khi B không còn phần tử, dừng thuật toán, trả về mảng kết quả D .

Mã giả thuật toán Non-Maximum Suppression:

```

procedure NMS( $B, S, N_t$ )
begin
     $D \leftarrow \emptyset$ 
    while  $B \neq \text{empty}$  do
         $m \leftarrow \text{argmax } S$ 
         $M \leftarrow b_m$ 
         $D \leftarrow D \cup M$ 
         $B \leftarrow B - M$ 
        for  $b_i$  in  $B$  do
            if  $\text{iou}(M, b_i) \geq N_t$  then
                 $B \leftarrow B - b_i$ 
                 $S \leftarrow S - s_i$ 
            end
        end
    end

```

```

return D,S
end

```

Việc chọn ngưỡng trùng lặp N là yếu tố quan trọng ảnh hưởng đến giá trị trả ra của hàm số và kết quả cuối cùng của mô hình SSD. Việc gán một giá trị ngưỡng cố định không hiệu quả. Khi một bounding box thuộc một đối tượng khác, nhưng lại quá trùng lặp với một đối tượng khác, bounding box này sẽ bị xóa. Điều này sẽ làm giảm độ chính xác của mô hình nhận diện vật thể.

Hàm số Soft-NMS được phát triển nhằm khắc phục ngưỡng cố định của hàm NMS, với ý tưởng là thay vì xóa hoàn toàn bounding box có độ trùng lặp lớn hơn ngưỡng, hàm Soft-NMS sẽ giảm độ tự tin của bounding box đó bằng một lượng trùng lặp. Nếu bounding box đó có giá trị độ tự tin cao, bounding box đó sẽ vẫn được giữ lại, như vậy đối tượng này sẽ không bị xóa.

Với bounding box thứ i có độ tự tin s_i , giá trị bounding box có độ tự tin lớn nhất M và ngưỡng trùng lặp N , ta có

$$s_i = \begin{cases} s_i, & IoU(M, b_i) < N \\ s_i(1 - IoU(M, b_i)), & IoU(M, b_i) \geq N \end{cases}$$

Mã giả thuật toán Soft Non-Maximum Suppression:

```

procedure Soft_NMS(B, S, Nt)
begin
    D ← ∅
    while B ≠ empty do
        m ← argmax S
        M ← bm
        D ← D ∪ M
        B ← B − M
        for bi in B do
            si ← si f(iou(M, bi))
        end
    end
    return D,S
end

```

3.5.3. Hàm Số Hard Negative Mining

Vì số lượng default box lớn hơn đáng kể so với số lượng ground truth box, do đó trong quá trình huấn luyện, hầu hết các default box sẽ có giá trị trùng lặp (*IoU*) với ground truth box thấp, như vậy các default box này sẽ được đánh giá là tiêu cực và được đánh nhãn là “background”. Như vậy tỉ lệ giữa default box tiêu cực sẽ nhiều hơn default box tích cực, và số lượng nhãn “background” chiếm ưu thế. Điều này tạo ra một sự không cân bằng trong quá trình huấn luyện. Khi đó mô hình sẽ chỉ học được “background”.

Thay vì sử dụng tất cả các default box tiêu cực, mô hình SSD áp dụng thuật toán Hard Negative Mining, với ý tưởng là sắp xếp các default box này theo độ tự tin giảm dần, và giới hạn tỉ lệ giữa số lượng default box tiêu cực và default box tích cực ở ngưỡng nhất định. Điều này làm giảm số lượng tính toán cũng như tăng hiệu quả của quá trình huấn luyện.

3.5.4. Hàm Số Tính Toán Lỗi Loss Function

Hàm số tính toán lỗi trong mô hình nhận diện vật thể SSD gồm hai thành phần là hàm lỗi của module dự đoán vị trí (localization loss) và module phân loại vật thể (confidence loss).

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

Với N là số lượng default box trùng khớp sau khi qua hai hàm số NMS và Hard Negative Mining. Nếu $N = 0$, ta có hàm số lỗi có giá trị bằng 0. Giá trị trọng số của hàm lỗi module dự đoán vị trí α được thiết lập bằng 1.

Giá trị của hàm lỗi dự đoán vị trí được tính trên hàm số lỗi Smooth L1 giữa giá trị box dự đoán (l) và giá trị ground truth box (g). Giá trị hàm lỗi được tính bằng giá trị hồi quy của thông tin chênh lệch của điểm trung tâm (cx, cy) của default box (d) và giá trị chiều dài (w) và chiều rộng (h) của default box này. Với x_{ij}^p là độ trùng khớp giữa default box thứ i và ground truth box thứ j cho lớp thứ p , và $\sum_i x_{ij}^p \geq 1$.

$$\begin{aligned} L_{loc}(x, l, g) &= \sum_{i \in Pos}^N \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m) \\ \hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx})/d_i^w \\ \hat{g}_j^{cy} &= (g_j^{cy} - d_i^{cy})/d_i^h \end{aligned}$$

$$\hat{g}_j^w = \log \left(\frac{g_j^w}{d_i^w} \right)$$

$$\hat{g}_j^h = \log \left(\frac{g_j^h}{d_i^h} \right)$$

Giá trị hàm lỗi phân loại vật thể được tính bằng giá trị softmax thông qua độ tự tin (c).

$$L_{conf}(x, c) = - \sum_{i \in Pos}^N x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \text{ where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

3.6. Thực Nghiệm Mô Hình SSD [18]

3.6.1. Kết Quả Thực Nghiệm Trên Tập Dữ Liệu PASCAL VOC

Theo bài nghiên cứu, nhóm tác giả so sánh giữa mô hình SSD với Fast R-CNN, Faster R-CNN và YOLO trên tập ảnh kiểm thử của tập dữ liệu VOC2007 và VOC2012. Các mô hình được tùy chỉnh trên mô hình tích chập VGG16 và có cấu hình tương tự nhau.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
			74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07	66.9	70.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	76.2	57.6	87.3	88.2	88.6	60.5	85.4	76.7	87.5	89.2	84.5	81.4	55.0	81.9	81.5	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	81.6	86.6	88.3	82.4	76.0	66.3	88.6	88.9	89.1	65.1	88.4	73.6	86.5	88.9	85.3	84.6	59.1	85.0	80.4	87.4	81.2

Hình 3.8 Kết Quả Thực Nghiệm Trên Tập Dữ Liệu PASCAL VOC2007 [18]

Kết quả thực nghiệm trên tập VOC2007 cho thấy với dữ liệu đầu vào kích thước nhỏ 300×300 , mô hình SSD300 đã có độ chính xác cao hơn Fast R-CNN. Khi kiểm thử với mô hình SSD512, kết quả cho thấy mô hình SSD512 cao hơn Faster R-CNN 1.7% mAP. Khi huấn luyện mô hình SSD với bộ dữ liệu VOC2007 và VOC2012, kết quả thực nghiệm SSD300 tốt hơn Faster R-CNN 1.1% mAP và SSD512 tốt hơn Faster R-CNN 3.6%. Kết quả tốt nhất trên tập kiểm thử của tập dữ liệu VOC2007 đạt được với mô hình SSD512 với dữ liệu đầu vào kích thước 512×512 , huấn luyện trên tập dữ liệu PASCAL VOC2007, PASCAL VOC2012 và COCO, với mAP đạt 81.6% mAP.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[2]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	74.1	88.4	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	80.0	90.7	86.8	80.5	67.8	60.8	86.3	85.5	93.5	63.2	85.7	64.4	90.9	89.0	88.9	86.8	57.2	85.1	72.8	88.4	75.9

Hình 3.9 Kết Quả Thực Nghiệm Trên Tập Dữ Liệu PASCAL VOC2012 [18]

Kết quả thực nghiệm khi huấn luyện với tập dữ liệu VOC2012 trainval, VOC2007 trainval và VOC2007 test (21503 ảnh dữ liệu đầu vào), kiểm thử với tập dữ liệu VOC2012 test (10991 ảnh dữ liệu đầu vào). Kết quả cho thấy mô hình SSD300 có cải thiện độ chính xác đáng kể hơn Fast R-CNN và Faster R-CNN. Độ chính xác của mô hình SSD512 vượt trội hơn Faster R-CNN với 4.5% mAP. Kết quả tốt nhất trên tập kiểm thử của tập dữ liệu VOC2012 đạt được với mô hình SSD512 với dữ liệu đầu vào kích thước 512×512 , huấn luyện trên tập dữ liệu PASCAL VOC2007 trainval, PASCAL VOC2007 test, PASCAL VOC2012 trainval và COCO, với mAP đạt 80.0% mAP.

3.6.2. Thời Gian Thực Thi Của Mô Hình SSD

Trong quá trình thực thi, với số lượng lớn default box được tạo ra, hàm số Non-Maximum Suppression đã hiệu quả trong việc giảm thiểu tính toán và tối ưu thời gian thực thi, phù hợp cho các tác vụ nhận diện vật thể theo thời gian thực. Khi thực nghiệm với vi xử lý Intel Xeon E5-2667 v3 @ 3.2GHz, bộ xử lý đồ họa Titan X, cuDNN v4 và giá trị batchsize 8, mô hình SSD chỉ sử dụng những box có độ tự tin lớn hơn 0.01, sử dụng hàm số NMS với hệ số trùng lặp Jaccard 0.45 và chỉ lấy 200 box có độ tự tin lớn nhất, tổng số lớp cần phân loại là 20 lớp, thời gian thực thi đo được trên mô hình SSD300 xấp xỉ 1.7 mili giây.

Khi so sánh giữa SSD, Faster R-CNN và YOLO, mô hình SSD300 và SSD512 đều đạt thời gian thực thi tối ưu và độ chính xác hơn Faster R-CNN. Mặc dù phiên bản Fast YOLO đạt được 155 khung hình trên giây, tuy nhiên độ chính xác lại giảm 22%. SSD300 vẫn được đánh giá là mô hình nhận diện vật thể thời gian thực đạt được độ chính xác trên 70% mAP. 80% thời gian thực thi của SSD dành cho quá trình lan truyền tiến của mạng tích chập trích xuất đặc trưng (mô hình VGG-16), nên khi sử dụng mô

hình mạng tích chập có thời gian thực thi ngắn hơn, tốc độ của mô hình SSD sẽ được cải thiện

3.6.3. Các yếu tố ảnh hưởng đến hiệu suất của mô hình SSD

- Tăng cường dữ liệu

Mô hình Fast R-CNN và Faster R-CNN sử dụng dữ liệu đầu vào và phép biến đổi lật ngang (horizontal flip) cho dữ liệu huấn luyện. Khi áp dụng các kỹ thuật tăng cường dữ liệu, mô hình SSD cải thiện 8.8% mAP.

- Số lượng Default Box

Như đã trình bày ở phần 3.4, số lượng default box được xác định ở mỗi vị trí được tạo ra là 6. Khi loại bỏ hai box có tỉ lệ 3 và 1/3, hiệu suất giảm 0.6%, khi loại bỏ thêm hai box có tỉ lệ 2 và 1/2, hiệu suất giảm thêm 2.1%. Như vậy, bằng việc sử dụng nhiều default box với đa dạng tỉ lệ, hiệu suất mô hình được cải thiện hơn.

- Sử dụng lớp tích chập Atrous

Khi sử dụng lớp tích chập Atrous của mạng trích xuất đặc trưng VGG-16, hiệu suất của mô hình cải thiện không đáng kể, tuy nhiên tốc độ xử lý được cải thiện đáng kể. Khi sử dụng mạng trích xuất đặc trưng VGG-16 hoàn chỉnh, kết quả tương tự nhưng tốc độ chậm hơn 20%.

Bảng 3.2 Các Thành Phần Ảnh Hưởng Đến Hiệu Suất Mô Hình SSD [18]

	SSD 300				
Sử dụng tăng cường dữ liệu?		✓	✓	✓	✓
Sử dụng default box tỉ lệ $\left\{\frac{1}{2}, 2\right\}$?	✓		✓	✓	✓
Sử dụng default box tỉ lệ $\left\{\frac{1}{3}, 3\right\}$?	✓			✓	✓
Sử dụng lớp tích chập Atrous?	✓	✓	✓		✓
Độ chính xác mAP trên VOC2007 test	65.5	71.6	73.7	74.2	74.3

- Số lượng ma trận trích xuất đặc trưng

Khi thử nghiệm bằng cách loại bỏ bớt các ma trận đặc trưng trong 6 ma trận đặc trưng ban đầu, với các điều kiện kiểm tra tương tự. Khi loại bỏ các ma trận đặc trưng, nhóm tác giả cố gắng tạo ra số lượng default box gần bằng số lượng ban đầu (8732 default box). Kết quả cho thấy việc giảm độ chính xác đáng kể khi sử dụng ít ma trận

đặc trưng, độ chính xác giảm từ 74.3% xuống 62.4% khi chỉ sử dụng một ma trận trích xuất đặc trưng.

Bảng 3.3 Ảnh Hưởng Của Số Lượng Ma Trận Trích Xuất Đặc Trưng [18]

Giá Trị Dự Đoán Từ Ma Trận Đặc Trưng						Độ chính xác mAP	Số lượng default box	
conv4	conv7	conv8	conv9	conv10	conv11	Sử dụng defaul box		
✓	✓	✓	✓	✓	✓	74.3	63.4	8732
✓	✓	✓	✓	✓		74.6	63.1	8764
✓	✓	✓	✓			73.8	68.4	8942
✓	✓	✓				70.7	69.2	9864
✓	✓					64.2	64.4	9025
	✓					62.4	64.0	8664

4. CHƯƠNG 4. TRIỂN KHAI MÔ HÌNH NHẬN DIỆN PHƯƠNG TIỆN GIAO THÔNG VIỆT NAM

4.1. Cài Đặt Mô Hình Nhận Diện Phương Tiện Giao Thông Việt Nam

Yêu cầu cài đặt: Ngôn ngữ Python từ phiên bản 3.6.0 trở lên.

Khuyến khích cài đặt với nền tảng Anaconda hoặc MiniAnaconda.

4.1.1. Cài Đặt Mô Hình Nhận Diện Phương Tiện Giao Thông:

Khởi tạo môi trường trên Anaconda (khuyến khích):

```
conda create --name vehicle_env  
conda activate vehicle_env
```

Tải mã nguồn tại https://github.com/LeNguyenGiaBao/vehicle_detection hoặc sử dụng lệnh git (đã cài đặt Git) như sau:

```
git clone https://github.com/LeNguyenGiaBao/vehicle_detection.git
```

Cài đặt các thư viện cần thiết

```
cd vehicle_detection  
pip install -r requirements.txt
```

4.1.2. Thực Thi Chương Trình Với Ảnh Phương Tiện Giao Thông

Thực thi chương trình thực thi trên một ảnh: với “path_of_image” là đường dẫn tương đối hoặc đường dẫn tuyệt đối của ảnh thực thi.

```
python inference.py -input=path_of_image
```

4.1.3. Triển Khai Website

Triển khai website:

```
streamlit run app.py
```

Website được thực thi tại đường dẫn <http://localhost:8501>

4.1.4. Triển Khai Ứng Dụng Cảnh Báo Lưu Lượng Giao Thông Qua Tin Nhắn

Thực thi lệnh:

```
python monitor_traffic.py
```

Kênh tin nhắn tại: <https://app.slack.com/client/T02RPGAG9D5/C02RXEK7806>

4.1.5. Huấn Luyện Mô Hình Nhận Diện

Cấu trúc tập dữ liệu theo chuẩn VOC như sau:

- Thư mục “Annotations”: chứa nhãn dữ liệu theo định dạng “.xml”
- Thư mục “ImageSets/Main”: chứa 2 tệp “train.txt” và “val.txt”, mỗi tệp chứa tên tệp dữ liệu.
- Thư mục “JPEGImages”: chứa dữ liệu hình ảnh.
- Tệp “labels.txt”: chứa tên các lớp dữ liệu, phân cách bởi dấu phẩy.

Thay đổi đường dẫn tập dữ liệu ở biến “dataset_path”, dòng 90, tệp train.py.

Thực thi chương trình huấn luyện:

```
python train.py
```

4.2. Quá Trình Huấn Luyện

Với tập dữ liệu phương tiện giao thông Việt Nam, gồm 2586 ảnh, nhóm đã chia thành hai tập dữ liệu: dữ liệu huấn luyện (gồm 2000 ảnh) và dữ liệu kiểm thử (gồm 586 ảnh). Mô hình nhóm sử dụng là SSD300 với mạng trích xuất đặc trưng VGG-16, số lượng default box là 8732. Mô hình huấn luyện trên vi xử lý Intel Xeon E5-2678 v3 @ 3.1 GHz, bộ xử lý đồ họa Intel Geforce GTX 3060, cuDNN v8 và giá trị batchsize 24, trong 8 giờ huấn luyện. Mô hình đạt được kết quả tốt nhất với giá trị hàm lỗi nhỏ nhất đạt 1.899 ở epoch thứ 170.

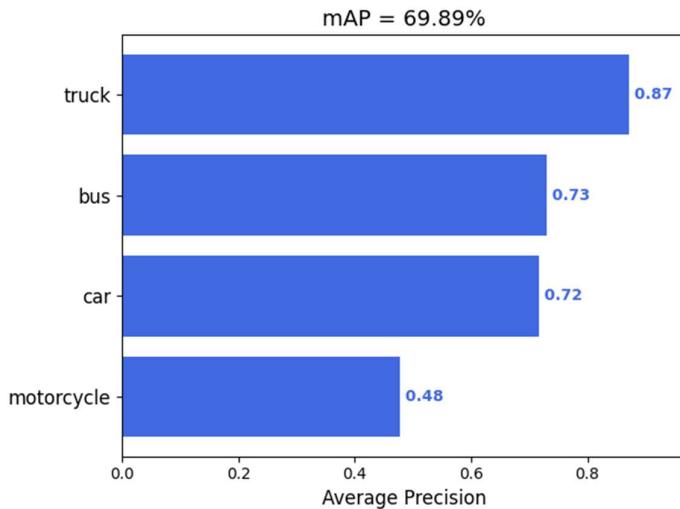
4.3. Kết Quả Thực Tế Trên Tập Dữ Liệu Giao Thông Việt Nam

Độ chính xác của mô hình trên tập dữ liệu phương tiện giao thông Việt Nam, với bốn lớp, độ chính xác của từng trên thang đo AP như sau:

Giá trị độ chính xác trung bình mAP của mô hình đạt được 69.89%.

Bảng 4.1 Độ Chính Xác Của Mô Hình Trên Tập Dữ Liệu Giao Thông Việt Nam

Lớp	motorcycle	car	bus	truck
Độ chính xác mAP	0.48	0.72	0.73	0.87



Hình 4.1 Độ Chính Xác Của Mô Hình Trên Tập Dữ Liệu Giao Thông Việt Nam

Độ chính xác của mô hình SSD trên các vật thể thuộc lớp “motorcycle” do những nguyên nhân sau:

- Về kiến trúc mô hình SSD: giá trị dữ liệu đầu vào có kích thước 300×300 , nhưng ma trận trích xuất đặc trưng đầu tiên có kích thước 38×38 . Tỉ lệ kích thước giảm khiến cho các vật thể nhỏ như xe máy bị mất thông tin từ dữ liệu đầu vào.
- Về dữ liệu thực tế: do góc camera ở xa vật thể, chất lượng camera không tốt, ảnh hưởng đến chất lượng hình ảnh của vật thể xe máy. Khi đó kích thước của vật thể sẽ rất nhỏ và không thể hiện rõ đặc trưng của vật thể.
- Về tăng cường dữ liệu: như đã đề cập ở trên, khi dữ liệu của vật thể xe máy nhỏ, các kỹ thuật tăng cường dữ liệu không thể hiện hiệu quả.
- Về văn hóa giao thông, tỷ lệ xe máy chiếm phần lớn, mật độ giao thông cao, dẫn tới việc trùng lặp giữa các vật thể.

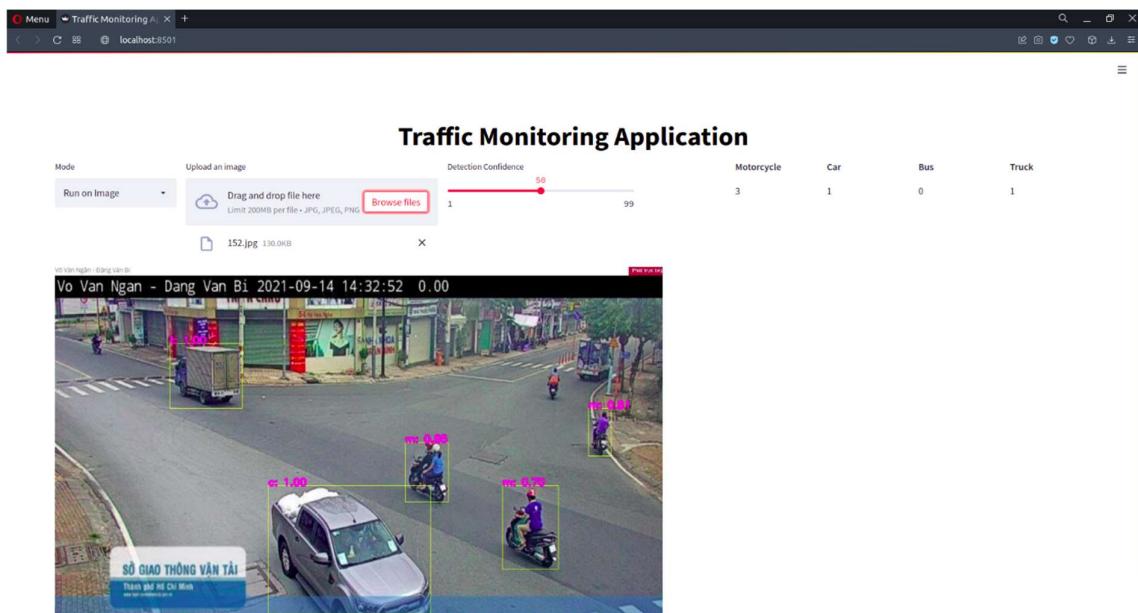
Để cải thiện độ chính xác trên vật thể thuộc lớp “motorcycle”, nhóm đã tiến hành thu thập thêm dữ liệu xe máy ở giai đoạn 2, kết quả cải thiện sau khi tăng số lượng dữ liệu.

Khi thực nghiệm với vi xử lý Intel Core i5 8265U @ 1.6GHz, bộ xử lý đồ họa Geforce MX230, cuDNN v8 và giá trị batchsize 1, thời gian thực thi trên một tấm ảnh đạt được ở mức 0.1 giây và thời gian thực thi trên tập dữ liệu kiểm thử đạt được trung bình 0.12 giây.

4.4. Triển Khai Website Nhận Diện Phương Tiện Giao Thông

Để trực quan kết quả trả về từ mô hình nhận diện vật thể, nhóm đã triển khai mô hình trên nền tảng web với thư viện Streamlit. Streamlit là một thư viện Python mã nguồn mở, dùng để xây dựng những website trực quan hóa dữ liệu một cách đơn giản. Sử dụng ngôn ngữ Python nên Streamlit rất phù hợp với các dự án được triển khai bằng ngôn ngữ Python.

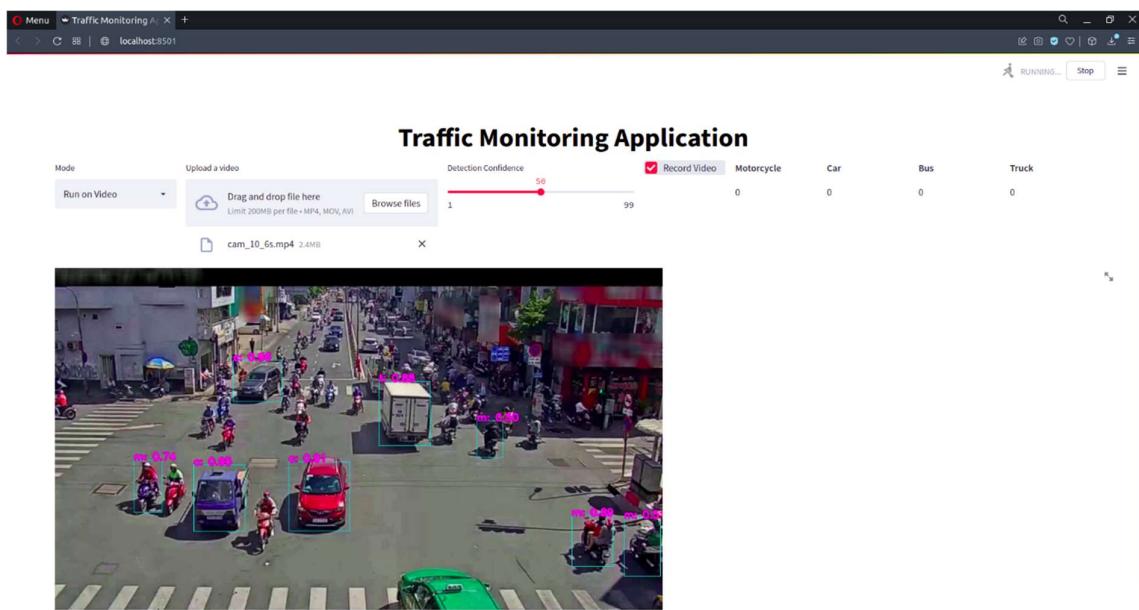
Chức năng chính của website là nhận diện phương tiện giao thông và đưa ra các thông tin liên quan như số lượng phương tiện, biểu đồ trực quan và đưa ra cảnh báo về tình trạng giao thông. Dữ liệu đầu vào có thể là ảnh, video hoặc trực tiếp từ hình ảnh camera của Cổng Thông tin Giao thông Thành phố Hồ Chí Minh.



Hình 4.2 Giao Diện Website Với Hình Ảnh Đầu Vào

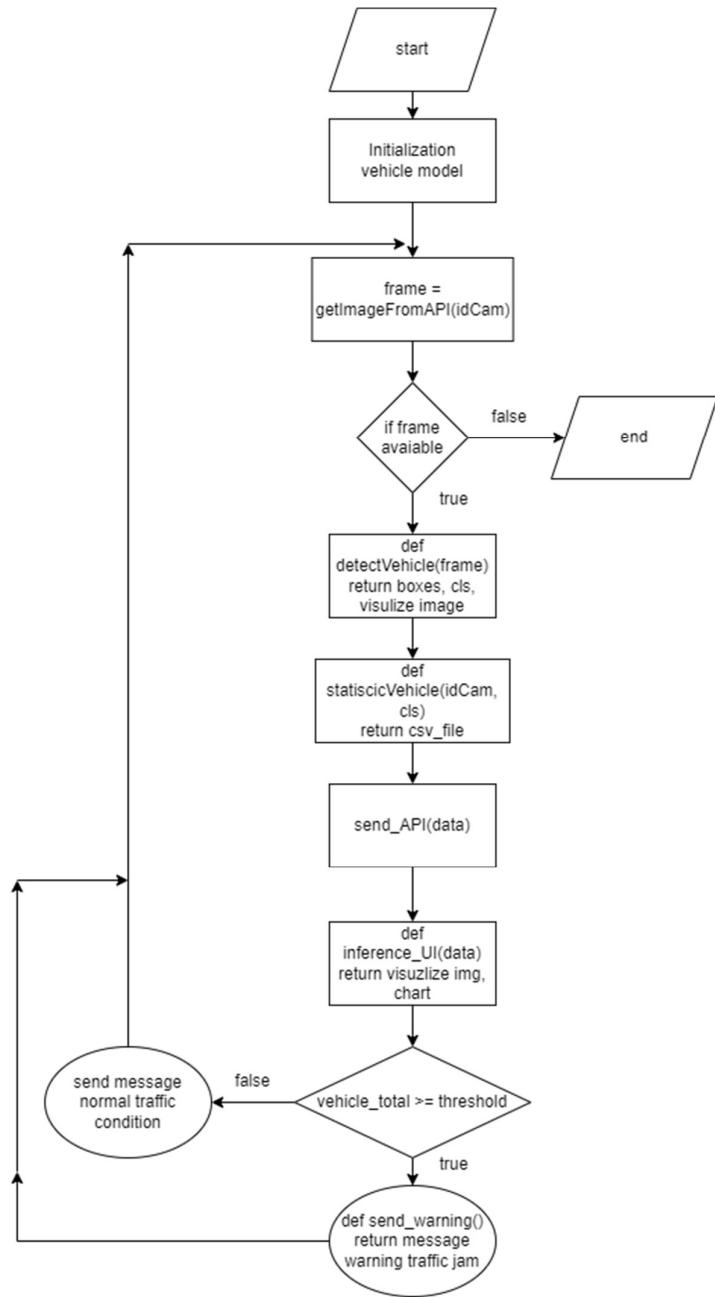


Hình 4.3 Giao Diện Website Với Hình Ảnh Tiếp Tự Camera Giao Thông



Hình 4.4 Giao Diện Website VỚI VIDEO ĐẦU VÀO

Luồng hoạt động với chức năng nhận diện vật thể từ hình ảnh trực tiếp từ camera giao thông.



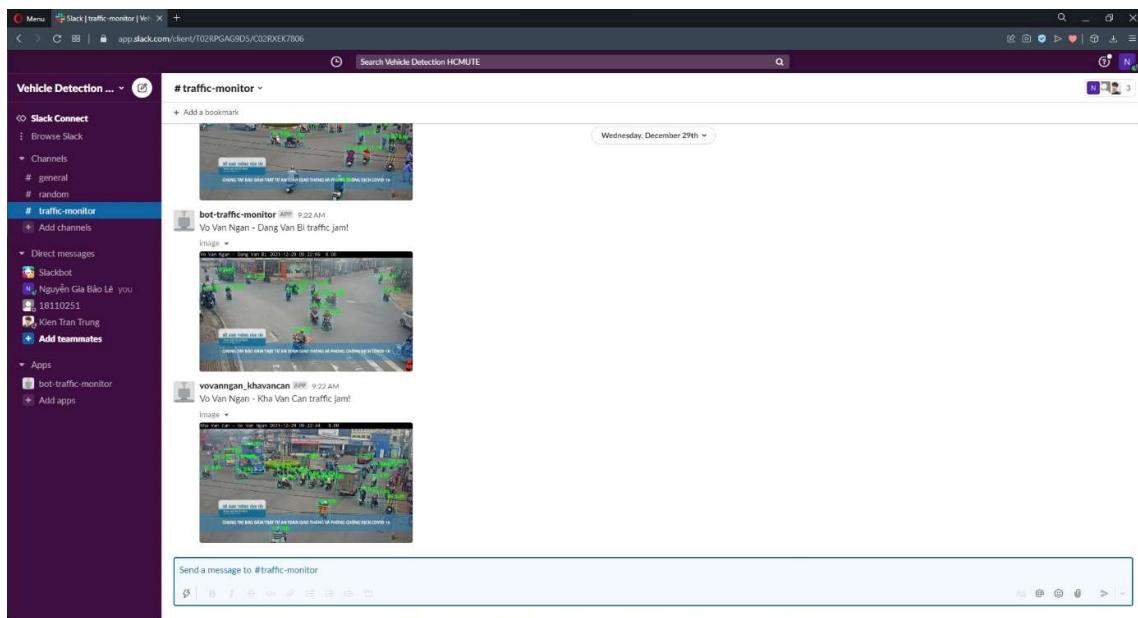
Hình 4.5 Luồng Hoạt Động Của Chức Năng Nhận Diện Từ Dữ Liệu Trực Tiếp

4.5. Triển Khai Kênh Tin Nhắn Cảnh Báo Lưu Lượng Giao Thông

Việc đưa ra thông tin cảnh báo về các vị trí có mật độ giao thông cao cũng như có nguy cơ ùn tắc giao thông sẽ tối ưu việc quản lý giao thông thông qua camera. Thông tin cảnh báo này cũng giúp truyền tải thông tin nhanh chóng đến các lực lượng chức năng làm nhiệm vụ, có thể tiếp cận vị trí nhanh chóng, giảm thiểu ùn tắc giao thông.

Nhóm triển khai việc gửi tin nhắn thông qua trang tin nhắn Slack, với các ưu điểm như có thể gửi tin nhắn, hình ảnh, ... một cách thuận tiện, đơn giản. Slack cũng cho phép tích hợp nhiều tiện ích và công cụ hữu ích cho công việc.

Nhóm sử dụng Slack Bots API, việc triển khai thông qua ngôn ngữ lập trình Python đơn giản, việc cấu hình đơn giản. Thông tin tin nhắn cảnh báo gồm tên địa điểm có nguy cơ tắc giao thông, và hình ảnh từ camera đó. Như vậy các nhân viên quản lý và các lực lượng chức năng có thể truy xuất, kiểm tra và đưa ra đánh giá kịp thời, nhanh chóng.



Hình 4.6 Hình Ảnh Kênh Tin Nhắn Cảnh Báo Lưu Lượng Giao Thông

PHẦN KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC

Sau một thời gian nghiên cứu và thực hiện đề tài “Phân Tích Tình Trạng Giao Thông Qua Theo Dõi Lưu Lượng Xe Trên Đường”, nhóm chúng em đã đạt được những kết quả sau:

1.1. Kiến Thức Tìm Hiểu Được

Tìm hiểu được các khái niệm và ý tưởng trong bài toán nhận diện vật thể, các kiến trúc mạng và các mô hình tiêu biểu trong bài toán nhận diện vật thể.

Tìm hiểu được cách thức hoạt động, các hàm huấn luyện và hàm tối ưu trong mô hình nhận diện vật thể SSD. Nắm bắt được các ưu điểm, nhược điểm, các phương pháp cải thiện độ chính xác của mô hình SSD.

1.2. Chương Trình Thực Hiện

Thực hiện cài đặt và huấn luyện mô hình SSD với bộ dữ liệu phương tiện giao thông Việt Nam.

Xây dựng website để trực quan kết quả từ mô hình SSD với những chức năng cơ bản như sau:

- Nhận diện phương tiện từ ảnh đầu vào.
- Nhận diện phương tiện từ video đầu vào.
- Nhận diện phương tiện từ hình ảnh Công Thông Tin Giao Thông Thành Phố Hồ Chí Minh.

Triển khai kênh tin nhắn cảnh báo lưu lượng giao thông khi mật độ giao thông cao, có nguy cơ ùn tắc giao thông.

2. ƯU ĐIỂM

- Nhận diện các vật thể đạt độ chính xác tốt.
- Thời gian thực thi nhanh, đạt được mức thực thi theo thời gian thực.
- Trực quan hóa mô hình trên nền tảng website trực quan, dễ sử dụng.

3. NHƯỢC ĐIỂM

- Nhận diện với đối tượng xe máy đạt kết quả chưa tốt.
- Chưa nhận diện các phương tiện trong điều kiện ánh sáng không tốt (điều kiện ngược sáng, bóng râm tối, ban đêm, ...).

4. HƯỚNG PHÁT TRIỂN

- Cải thiện độ chính xác của mô hình, thời gian thực thi.
- Thực hiện nhận diện trong điều kiện ánh sáng không tốt.
- Xây dựng các chức năng nâng cao cho ứng dụng website như theo dõi đồng thời nhiều camera, hiển thị danh sách camera dạng bản đồ, ...

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Văn Minh, "Thành Phố Hồ Chí Minh Mỗi Ngày Hơn 750 Xe Ô Tô Và Mô Tô Đăng Ký Mới," *Trang Tin Điện Tử Đảng Bộ Thành Phố Hồ Chí Minh*, 2020.
- [2] "Công Thông Tin Giao Thông Thành Phố Hồ Chí Minh," Sở Giao Thông Vận Tải TP Hồ Chí Minh, [Trực tuyến]. Available: <http://giaothong.hochiminhcity.gov.vn>. [Đã truy cập 31/12/2021].
- [3] Vương Xuân Can, Phan Xuân Vũ, M. Rui-Fang, Vũ Trọng Thuật, Vũ Văn Duy và Nguyễn Duy Nội, "Vehicle Detection And Counting Under Mixed Traffic Conditions In Vietnam Using YoloV4," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 12-2-2021.
- [4] Đinh Đức Liêm, Nguyễn Hồng Nam, Thái Huy Tân và Lê Kim Hùng, "Towards AI-Based Traffic Counting System with Edge Computing," *Hindawi*, 28-1-2021.
- [5] X. Han, J. Chang và K. Wang, "Real-time Object Detection Based On YoloV2 For Tiny Vehicle Object," *ScienceDirect*, 2021.
- [6] A. Arinaldi, J. A. Pradana and A. A. Gurusinga, "Detection And Classification Of Vehicles For Traffic Video Analytics," *ScienceDirect*, 2018.
- [7] "Artificial Neural Network," *Wikipedia*.
- [8] Vũ Hữu Tiệp, "Logistic Regression," *Machine Learning cơ bản*, 2017.
- [9] Vũ Hữu Tiệp, "Multi-layer Perceptron và Backpropagation," *Machine Learning Cơ Bản*, 2017.
- [10] K. Sarkar, "ReLU: Not A Differentiable Function. Why Used In Gradient Based Optimization," *Medium*, 2018.
- [11] A. Zhang, Z. Lipton, M. Li và A. J. Smola, "Convolutional Neural Networks - Convolutions for Images," *Dive Into Deep Learning*.
- [12] A. Zhang, Z. Lipton, M. Li và A. J. Smola, "Convolutional Neural Networks - Padding and Stride," *Dive Into Deep Learning*.
- [13] A. Zhang, Z. Lipton, M. Li và A. J. Smola, "Convolutional Neural Networks - Pooling," *Dive Into Deep Learning*.
- [14] A. Zhang, Z. Lipton, M. Li và A. J. Smola, "Modern Convolutional Neural Network," *Dive Into Deep Learning*.
- [15] "ImageNet," *Wikipedia*, 2021.
- [16] J. Jordan, "An Overview Of Object Detection: One-Stage Methods," *Jeremy Jordan*, 2018.

- [17] S. Park, "A Guide To Two-Stage Object Detection: R-CNN, FPN, Mask R-CNN," *Medium*, 2021.
- [18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu và A. C. Berg, "SSD: Single Shot MultiBox Detector," *arXiv*, 2016.
- [19] K. Simonyan và A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, 2015.
- [20] "Image Classification on ImageNet," *Papers With Code*, 2021.
- [21] "Jaccard Index," *Wikipedia*, 2021.
- [22] D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, "Scalable Object Detection Using Deep Neural Networks," 2013.

PHỤ LỤC

PHỤ LỤC 1. HỆ SỐ JACCARD [21].....	59
PHỤ LỤC 2. MỘT SỐ HÌNH ẢNH TỪ CAMERA GIAO THÔNG.....	60
PHỤ LỤC 3 BIỂU ĐỒ LUU LUONG PHUONG TIEN GIAO THÔNG TAI NUT GIAO VÕ VĂN NGÂN – ĐẶNG VĂN BI NGÀY 28/12/2021	62

PHỤ LỤC 1. HỆ SỐ JACCARD [21]

Hệ số Jaccard (Jaccard similarity coefficient) được phát triển bởi Paul Jaccard và Tanimoto. Hệ số Jaccard là một phép toán thống kê giữa độ tương đồng của hai tập hợp mẫu. Hệ số Jaccard đo lường sự giống nhau giữa hữu hạn các tập mẫu, giá trị dựa trên thương số của phần giao (intersection) và phần hợp (union) của các tập mẫu.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Miền giá trị của hệ số Jaccard trong đoạn $[0, 1]$. Khi cả A và B là tập rỗng, ta có $J(A, B) = 1$. Giá trị hệ số Jaccard càng cao thì độ tương đồng của hai tập mẫu càng cao. Khi hệ số Jaccard đạt giá trị 1, hai tập mẫu trùng nhau, tương tự khi hệ số Jaccard đạt giá trị 0, hai tập mẫu độc lập nhau.

Hệ số Jaccard được sử dụng rộng rãi trong nhiều ngành khoa học như khoa học máy tính, sinh học, địa chất học, ...

Khoảng cách Jaccard (Jaccard distance) được tính dựa trên độ khác biệt giữa hai hoặc nhiều tập mẫu, được tính bằng hiệu số giữa một và hệ số Jaccard, hoặc tính bằng thương số của hiệu số phần hợp và phần giao của các tập mẫu, với phần hợp của các tập mẫu. Khoảng cách Jaccard ứng dụng trong tính toán phân cụm hay giảm chiều dữ liệu

$$d_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

PHỤ LỤC 2. MỘT SỐ HÌNH ẢNH TỪ CAMERA GIAO THÔNG



Hình 1. Hình Ảnh Camera Nút Giao Võ Văn Ngân – Đặng Văn Bi



Hình 2. Hình Ảnh Camera Nút Giao Võ Văn Ngân – Kha Vạn Cân



Hình 3. Hình Ảnh Camera Nút Giao Trường Chinh – Tân Kỳ Tân Quý



Hình 4. Hình Ảnh Camera Nút Giao Lý Thường Kiệt – Lạc Long Quân



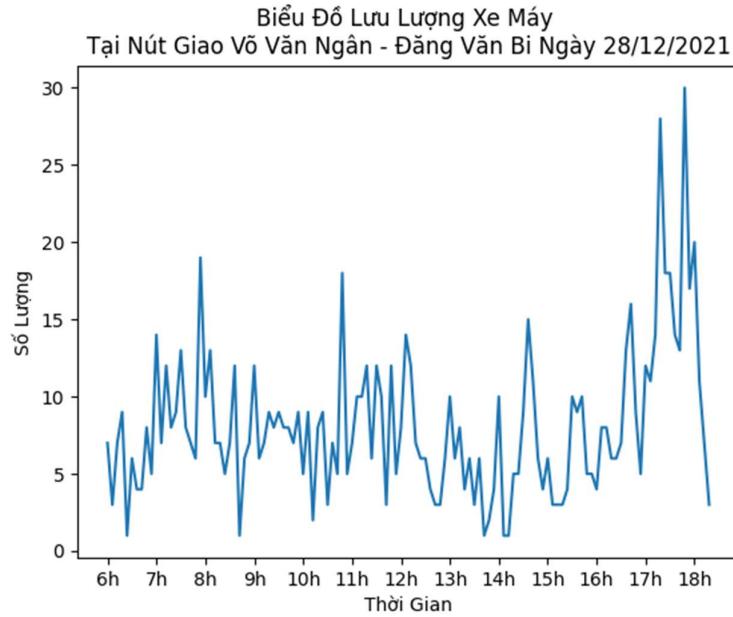
Hình 5. Hình Ảnh Camera Nút Giao Hoàng Văn Thụ – Trần Huy Liệu



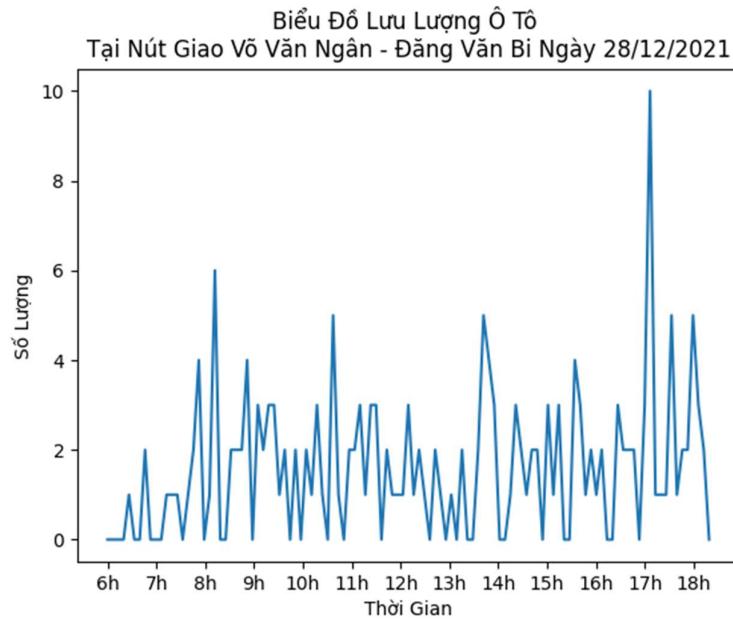
Hình 6. Hình Ảnh Camera Nút Giao Lê Quang Định – Nơ Trang Long

**PHỤ LỤC 3 BIỂU ĐỒ LUU LƯỢNG PHƯƠNG TIỆN GIAO THÔNG TẠI NÚT
GIAO VÕ VĂN NGÂN – ĐẶNG VĂN BI NGÀY 28/12/2021**

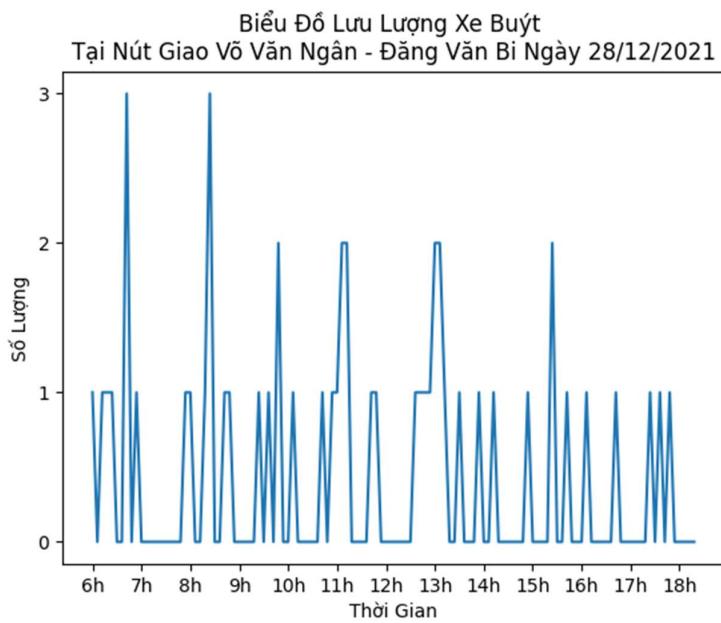
Dữ liệu lưu lượng phương tiện giao thông được ghi nhận tại nút giao Võ Văn Ngân – Đặng Văn Bi, từ 6 giờ 10 phút đến 18 giờ 30 phút ngày 28/12/2021.



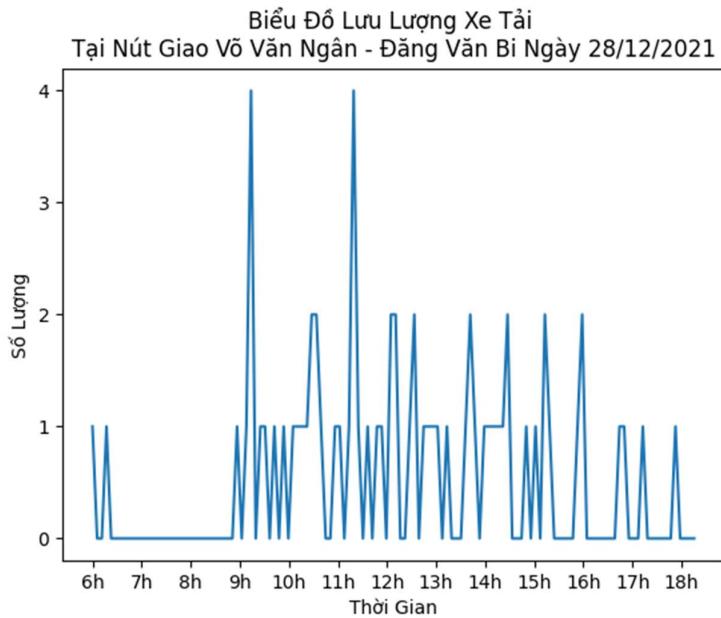
**Hình 7. Biểu Đồ Lưu Lượng Xe Máy Tại Nút Giao Võ Văn Ngân – Đặng Văn Bi Ngày
28/12/2021**



Hình 8. Biểu Đồ Lưu Lượng Ô Tô Tại Nút Giao Võ Văn Ngân – Đặng Văn Bi Ngày 28/12/2021



Hình 9. Biểu Đồ Lưu Lượng Xe Buýt Tại Nút Giao Võ Văn Ngân – Đặng Văn Bi Ngày 28/12/2021



Hình 10. Biểu Đồ Lưu Lượng Xe Tải Tại Nút Giao Võ Văn Ngân – Đặng Văn Bi Ngày 28/12/2021