

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

XÂY DỰNG HỆ THỐNG BÃI ĐỖ XE THÔNG
MINH ỨNG DỤNG THỊ GIÁC MÁY TÍNH TRONG
NHẬN DẠNG VÀ QUẢN LÝ PHƯƠNG TIỆN
DEVELOPING AN INTELLIGENT PARKING
SYSTEM USING COMPUTER VEHICLE
DETECTION AND MANAGEMENT

Sinh viên: Đào Việt Anh

Mã số: B2113303

Khóa: 47

Giảng viên hướng dẫn: ThS. Phạm Nguyên Hoàng

Cần Thơ, 12/2025

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

XÂY DỰNG HỆ THỐNG BÃI ĐỖ XE THÔNG
MINH ỨNG DỤNG THỊ GIÁC MÁY TÍNH TRONG
NHẬN DẠNG VÀ QUẢN LÝ PHƯƠNG TIỆN
DEVELOPING AN INTELLIGENT PARKING
SYSTEM USING COMPUTER VEHICLE
DETECTION AND MANAGEMENT

Giảng viên hướng dẫn:
ThS. Phạm Nguyên Hoàng

Sinh viên thực hiện:
Đào Việt Anh
Mã số: B2113303
Khóa: 47

Cần Thơ, 12/2025

TRƯỜNG ĐẠI HỌC CÀN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN
VÀ TRUYỀN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

XÁC NHẬN CHỈNH SỬA LUẬN VĂN
THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh): **XÂY DỰNG HỆ THỐNG BÃI
ĐỖ XE THÔNG MINH ÚNG DỤNG THỊ GIÁC MÁY TÍNH TRONG
NHẬN DẠNG VÀ QUẢN LÝ PHƯƠNG TIỆN (DEVELOPING AN
INTELLIGENT PARKING SYSTEM USING COMPUTER VEHICLE
DETECTION AND MANAGEMENT)**

Họ tên sinh viên: **Đào Việt Anh**

MSSV: **B2113303**

Mã lớp: **DI21Z6A1**

Đã báo cáo tại hội đồng ngành: **Khoa học máy tính**

Ngày báo cáo: 16/12/2025

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày tháng 12 năm 2025

Giáo viên hướng dẫn

(Ký và ghi họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN

Cần Thơ, ngày tháng năm

(GVHD ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn chân thành đến Quý Thầy Cô trong Trường CNTT&TT – Trường Đại học Cần Thơ đã tận tình giảng dạy, truyền đạt kiến thức quý báu trong suốt quá trình học tập và thực hiện đề tài.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến Thầy Phạm Nguyên Hoàng, người đã tận tình hướng dẫn, hỗ trợ em trong suốt quá trình nghiên cứu, thực hiện và hoàn thiện báo cáo này.

Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè và các anh/chị khóa trên đã luôn động viên, chia sẻ kinh nghiệm và hỗ trợ em trong suốt quá trình thực hiện đề tài.

Mặc dù đã có gắng hoàn thành đề tài với tinh thần nghiêm túc và cầu thị, nhưng chắc chắn vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, đánh giá của Quý Thầy Cô để em có thể hoàn thiện hơn trong những nghiên cứu sau này.

Em xin chân thành cảm ơn!

Cần Thơ, 12/2025

Người viết

Đào Việt Anh

MỤC LỤC

MỤC LỤC.....	1
DANH MỤC HÌNH	3
DANH MỤC TỪ VIẾT TẮT	4
ABSTRACT	5
TÓM TẮT	6
I.PHẦN GIỚI THIỆU	7
1.Đặt vấn đề	7
2.Những nghiên cứu liên quan	8
3.Mục tiêu đề tài.....	10
4.Phương pháp nghiên cứu	11
5.Bố cục	12
II.PHẦN NỘI DUNG	13
CHƯƠNG 1. MÔ TẢ BÀI TOÁN.....	13
1.1.Mô tả chi tiết bài toán	13
1.1.1.Nhận dạng logo phương tiện từ hình ảnh:	13
1.1.2.Nhận dạng và trích xuất tên phương tiện từ hình ảnh:	15
1.1.3.Xây dựng giao diện web hiển thị kết quả nhận dạng:	16
1.2.Vấn đề giải pháp liên quan đến bài toán.....	17
1.2.1.Công cụ lập trình – Visual Studio Code.....	17
1.2.2.Ngôn ngữ lập trình	17
1.2.3.Các giải pháp và thuật toán sử dụng trong hệ thống.	19
1.2.3.1.Giải pháp học sâu R-CNN (Region-based Convolutional Neural Network)	19
1.2.3.2.Giải pháp học sâu – CNN (Convolutional Neural Network)	20
1.2.3.3.Giải pháp phát hiện đối tượng – YOLOv8.....	20
1.2.3.3.1.Các chỉ số đánh giá mô hình: Precision, Recall, F1-score, CER:.....	21
1.2.3.4.Tách vùng văn bản – CRAFT	22
1.2.3.5.Giải pháp nhận diện chữ - OCR (Optical Character Recognition)	22
1.2.3.6.Công nghệ Fuzzy Matching	23
1.3.Cách đánh giá mô hình	27
CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP	28
2.1.Thiết kế hệ thống.....	28

Đề tài: Xây dựng ứng dụng web nhận diện và tìm kiếm hình ảnh thiên văn tích hợp chatbot

2.1.1.Sơ đồ tổng quát.....	28
2.1.2.Giao diện người dùng	29
2.2.Cài đặt giải pháp	32
2.2.1.Thu thập dữ liệu.....	32
2.2.2.Tiền xử lý dữ liệu	33
2.2.3.Huấn luyện mô hình nhận diện logo:.....	36
2.2.3.1.Quy trình huấn luyện mô hình R-CNN:.....	36
2.2.3.2.Quy trình huấn luyện mô hình CNN:	41
2.2.3.3.Quy trình huấn luyện mô hình YOLO:	44
2.2.4. Nhận diện tên xe:.....	47
2.2.4.1 Craft (Character-Region Awareness for Text detection):.....	47
2.2.4.2 OCR (Optical Character Recognition) :.....	48
2.2.5. Trả về kết quả thông qua fuzzy matching:	51
2.2.5.1. Thu thập dữ liệu để trả về:	51
2.2.5.2.Hệ thống Fuzzymather trả về kết quả chính xác:	52
CHƯƠNG 3. GIAO DIỆN , ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN	54
III.KẾT LUẬN.....	62
TÀI LIỆU THAM KHẢO	63

DANH MỤC HÌNH

Hình 2.1: Sơ đồ tổng quát của hệ thống.....	28
Hình 2.2: Dữ liệu thu thập	33
Hình 2.3: Dữ liệu đã qua kiểm tra	34
Hình 2.4: Cấu trúc dữ liệu để huấn luyện R-CNN.....	35
Hình 2.5: Ảnh sau khi tăng cường	36
Hình 2.6: Quá trình huấn luyện R-CNN lần 1	38
Hình 2.7: Quá trình huấn luyện R-CNN lần 2	39
Hình 2.8: Quá trình huấn luyện R-CNN lần 2	40
Hình 2.9: Quá trình huấn luyện R-CNN lần 4	41
Hình 2.10: Dữ liệu huấn luyện CNN.....	42
Hình 2.11: Quá trình huấn luyện CNN lần 1	42
Hình 2.12: Quá trình huấn luyện CNN lần 2	43
Hình 2.13: Kết quả của CNN	44
Hình 2.14: Kết quả lần 1 huấn luyện của YOLO.....	45
Hình 2.15: Kết quả lần 2 huấn luyện của YOLO.....	46
Hình 2.16: Kết quả huấn luyện của YOLO	46
Hình 2.17: Kết quả nhận diện của CRAFT	48
Hình 2.18: Kết quả so sánh các OCR.....	51
Hình 3.1: Giao diện nhận diện ảnh.....	54
Hình 3.2: Giao diện nhận kết quả.....	55
Hình 3.3: Giao diện nhận diện ảnh.....	55
Hình 3.4: Giao diện lịch sử đỗ xe.....	56
Hình 3.5: Giao diện quản lý	56
Hình 3.6: Giao diện tổng quát	57
Hình 3.7: Đánh giá và kiểm thử R-CNN	58
Hình 3.8: Đánh giá và kiểm thử CNN.....	59
Hình 3.9: Đánh giá và kiểm thử YOLO	60
Hình 3.11: Kiểm thử OCR	61
Hình 3.12: Kiểm thử Fuzzy matching	61

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt (Abbreviation)	Từ viết đầy đủ (Origin word)
1	AI	Artificial Intelligence
2	CNN	Convolutional Neural Network
3	R-CNN	Region-based Convolutional Neural Network
4	YOLO	You Only Look Once
5	CRAFT	Character Region Awareness for Text Detection
6	OCR	Optical Character Recognition
7	CSS	Cascading Style Sheets
8	HTML	HyperText Markup Language
9	JS	JavaScript
10	TrOCR	Transformer-based Optical Character Recognition
11	IoU	Intersection over Union
12	mAP	mean Average Precision
13	ReLU	Rectified Linear Unit
14	VGG	Visual Geometry Group

ABSTRACT

In the context of rapid urbanization, the demand for optimizing parking space and automating vehicle management processes has become increasingly essential. Stemming from this need, the project “Developing an Intelligent Parking System Applying Computer Vision for Vehicle Recognition and Management” was carried out to build a platform that supports users and administrators in automatically identifying vehicles, monitoring parking slot status, and conveniently and accurately retrieving related information through a web interface integrated with artificial intelligence.

The system is developed with key functionalities such as: recognizing license plates and classifying vehicle types using deep learning models, monitoring the status of parking spaces in real-time through cameras, searching and managing vehicle information based on license plates, entry-exit time, or specific features, and integrating a chatbot to assist with inquiries about parking regulations and system usage. The recognition model applies the YOLOv8 architecture combined with OCR for license plate extraction, trained on a preprocessed and annotated vehicle image dataset. The website is developed using common technologies such as PHP, JavaScript, and Python for image processing and artificial intelligence algorithms.

The results show that the system operates stably, achieving high accuracy in vehicle recognition, parking slot detection, and entry-exit management. This is a practical solution for smart parking lots, shopping centers, office buildings, as well as modern smart-city applications.

TÓM TẮT

Trong bối cảnh đô thị hóa phát triển nhanh chóng, nhu cầu tối ưu hóa không gian đỗ xe và tự động hóa các quy trình quản lý phương tiện ngày càng trở nên cấp thiết. Xuất phát từ nhu cầu đó, đề tài “Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính trong nhận dạng và quản lý phương tiện” được thực hiện nhằm phát triển một nền tảng hỗ trợ người dùng và nhà quản lý tự động nhận dạng phương tiện, giám sát vị trí đỗ xe và tra cứu thông tin liên quan một cách thuận tiện, chính xác thông qua giao diện web kết hợp trí tuệ nhân tạo.

Hệ thống được xây dựng với các chức năng chính như: nhận dạng biển số và phân loại loại phương tiện bằng mô hình học sâu (deep learning), giám sát tình trạng các vị trí đỗ xe theo thời gian thực thông qua camera, tìm kiếm và quản lý thông tin phương tiện theo biển số, thời gian vào – ra hoặc đặc điểm, đồng thời tích hợp chatbot hỗ trợ tương tác hỏi đáp về quy định đỗ xe và hướng dẫn sử dụng hệ thống. Mô hình nhận dạng ứng dụng kiến trúc YOLOv8 kết hợp OCR để trích xuất biển số, được huấn luyện trên tập dữ liệu hình ảnh phương tiện đã qua tiền xử lý và gán nhãn. Website được phát triển bằng các công nghệ phổ biến như PHP, JavaScript, và Python cho phần xử lý ảnh cùng các thuật toán trí tuệ nhân tạo.

Kết quả cho thấy hệ thống vận hành ổn định, đạt độ chính xác cao trong việc nhận dạng phương tiện, phát hiện trạng thái chỗ đỗ và quản lý thông tin ra vào. Đây là một giải pháp hữu ích cho các bãi đỗ xe thông minh, trung tâm thương mại, tòa nhà văn phòng, cũng như các ứng dụng đô thị thông minh hiện đại.

I.PHẦN GIỚI THIỆU

1.Đặt vấn đề

Trong bối cảnh đô thị hóa diễn ra nhanh chóng và số lượng phương tiện giao thông gia tăng mạnh mẽ, nhu cầu quản lý bãi đỗ xe một cách hiệu quả trở thành một trong những vấn đề quan trọng của các thành phố hiện đại. Từ lâu, việc tổ chức chỗ đỗ xe đã đóng vai trò thiết yếu trong hoạt động giao thông, an ninh cũng như vận hành các khu dân cư, trung tâm thương mại và cơ quan công cộng. Tuy nhiên, cùng với sự phát triển của hạ tầng giao thông và mật độ phương tiện, khối lượng dữ liệu liên quan đến hình ảnh phương tiện, biển số và tình trạng chỗ đỗ ngày càng trở nên lớn và phức tạp. Dữ liệu này không chỉ đa dạng về góc chụp, điều kiện ánh sáng, thời tiết mà còn khác nhau về chất lượng và định dạng, khiến việc quản lý, phân tích và xử lý tự động trở thành một thách thức đáng kể.

Trong lĩnh vực quản lý giao thông và bãi đỗ xe, hình ảnh đóng vai trò trung tâm, hỗ trợ việc giám sát, nhận dạng và theo dõi phương tiện. Tuy nhiên, để trích xuất thông tin từ các hình ảnh camera giám sát, nhân viên vận hành thường phải kiểm tra, ghi nhận và đổi chiếu thủ công, tốn nhiều thời gian và dễ xảy ra sai sót. Đối với những bãi đỗ xe lớn hoặc có mật độ phương tiện cao, khó khăn này càng tăng lên rõ rệt khi việc kiểm soát biển số, phát hiện vị trí còn trống hay theo dõi thời gian đỗ xe gần như không thể thực hiện hiệu quả nếu không có sự hỗ trợ của công nghệ. Hơn nữa, đối với người dùng phổ thông, việc tìm kiếm thông tin phương tiện, kiểm tra chỗ trống hoặc tra cứu lịch sử đỗ xe cũng không có nhiều công cụ hỗ trợ trực quan và thuận tiện.

Thực tế hiện nay cho thấy, dù đã tồn tại các hệ thống bãi đỗ xe bán tự động hay một số thiết bị nhận dạng biển số, nhưng chúng vẫn tồn tại nhiều hạn chế, bao gồm:

- Khó tiếp cận với người dùng phổ thông: giao diện phức tạp, thiếu tính trực quan.
- Thiếu khả năng nhận dạng trực tiếp từ hình ảnh: nhiều hệ thống chỉ ghi nhận thông tin, không hỗ trợ phân tích hình ảnh tự động.
- Không thống nhất về chuẩn dữ liệu: camera từ các nhà cung cấp khác nhau cho chất lượng và định dạng hình ảnh không đồng nhất.
- Gặp khó khăn trong môi trường thực tế: điều kiện ánh sáng yếu, mưa, ngược sáng khiến độ chính xác giảm.
- Hạn chế về tốc độ và độ chính xác trong việc nhận dạng biển số hoặc phát hiện vị trí đỗ xe còn trống.

Trong khi đó, trí tuệ nhân tạo (AI), đặc biệt là các mô hình học sâu (Deep

Learning) và mạng nơ-ron tích chập (CNN – Convolutional Neural Network), đã chứng minh hiệu quả vượt trội trong xử lý ảnh ở nhiều lĩnh vực như y tế, giao thông, an ninh... Công nghệ nhận dạng biển số xe (Automatic License Plate Recognition – ALPR) và phát hiện đối tượng (Object Detection) hiện nay đạt độ chính xác rất cao, hoạt động tốt ngay cả trong các điều kiện phức tạp. Tuy nhiên, việc áp dụng các công nghệ này vào xây dựng một hệ thống bãi đỗ xe thông minh tổng thể vẫn chưa được triển khai rộng rãi, đặc biệt là trong các môi trường thực tế tại Việt Nam, nơi hạ tầng camera và dữ liệu có nhiều biến động.

Do đó, việc nghiên cứu và xây dựng một giải pháp quản lý bãi đỗ xe thông minh ứng dụng thị giác máy tính là yêu cầu cấp thiết. Giải quyết tốt bài toán này không chỉ giúp tối ưu hóa quy trình quản lý, giảm tải nhân lực, mà còn góp phần nâng cao hiệu quả vận hành, tăng tính minh bạch và cải thiện trải nghiệm người dùng. Một hệ thống nhận dạng tự động có thể hỗ trợ phát hiện biển số, phân loại phương tiện, xác định vị trí còn trống, giám sát thời gian đỗ và tra cứu thông tin nhanh chóng, chính xác.

Đề tài mang lại nhiều giá trị thực tiễn:

- **Trong quản lý:** hỗ trợ tự động hóa ghi nhận phương tiện vào – ra, quản lý thời gian và chi phí đỗ xe, giảm sai sót trong quá trình vận hành.
- **Trong giám sát an ninh:** giúp nhận diện nhanh biển số, theo dõi phương tiện可疑 và lưu trữ dữ liệu phục vụ truy xuất khi cần thiết.
- **Đối với người dùng:** cung cấp công cụ tra cứu chỗ trống, kiểm tra thông tin phương tiện và hỗ trợ tìm kiếm một cách nhanh chóng, thuận tiện qua giao diện web thân thiện.
- **Đối với đô thị thông minh:** góp phần xây dựng hệ thống giao thông hiện đại, giảm ùn tắc và phục vụ quá trình số hóa hạ tầng đô thị.

2.Những nghiên cứu liên quan

Trong lĩnh vực ứng dụng trí tuệ nhân tạo vào giao thông thông minh, đã có nhiều công trình nghiên cứu liên quan đến việc nhận diện, phân loại và khai thác dữ liệu hình ảnh phương tiện.

Một số nghiên cứu tiêu biểu có thể kể đến như:

• **Zhang et al. (2019)** đã áp dụng CNN để phân loại phương tiện từ hình ảnh camera giám sát tại bãi đỗ xe. Mô hình đạt độ chính xác khoảng 95% trong việc phân biệt các loại xe như ô tô, xe máy, xe tải. Tuy nhiên, nghiên cứu chủ yếu tập trung vào dữ liệu chuyên dụng trong môi trường kiểm soát, chưa thân thiện với các hệ thống triển khai thực tế quy mô lớn.

• **Ball & Brunner (2010)** trình bày một khảo sát toàn diện về ứng dụng của học máy trong phân tích dữ liệu giao thông, bao gồm nhận dạng biển số, phân loại phương tiện, và phát hiện xe vi phạm. Công trình nhấn mạnh tiềm năng của các thuật toán tự

động trong quản lý giao thông, nhưng cũng chỉ ra rằng độ chính xác phụ thuộc đáng kể vào chất lượng và điều kiện của dữ liệu đầu vào.

• **Parking Radar (ứng dụng thương mại)**: Một số hệ thống hiện đại sử dụng cảm biến siêu âm hoặc camera để phát hiện chỗ trống trong bãi đỗ xe và hiển thị lên bản đồ cho người dùng. Tuy nhiên, các hệ thống này thường sử dụng dữ liệu tĩnh, không tích hợp AI, và không hỗ trợ nhận dạng biển số trực tiếp từ hình ảnh.

• **Parkopedia Web Platform**: Đây là nền tảng tìm kiếm bãi đỗ xe phổ biến trên thế giới, cung cấp bản đồ và thông tin chỗ đỗ. Tuy nhiên, hệ thống không tích hợp nhận dạng phương tiện, chỉ hoạt động dựa trên dữ liệu mà người dùng cung cấp trước.

• **Kim và cộng sự (2020)** xây dựng hệ thống nhận diện biển số xe sử dụng mạng học sâu trong môi trường đô thị. Hệ thống đạt độ chính xác khoảng 90%, nhưng chưa hỗ trợ các chức năng quản lý bãi đỗ xe, tra cứu dữ liệu hay giao diện web trực quan cho người dùng.

• **Wang & Xu (2021)** phát triển ứng dụng di động sử dụng công nghệ AR để chỉ hướng và vị trí bãi đỗ xe trong thực tế, giúp người dùng tìm được nơi đỗ xe gần nhất. Tuy nhiên, ứng dụng không tích hợp khả năng nhận dạng phương tiện từ hình ảnh.

• **Shamir et al. (2014)** xây dựng hệ thống tìm kiếm phương tiện trong cơ sở dữ liệu hàng triệu ảnh camera dựa trên đặc trưng hình dạng và màu sắc. Tuy nhiên, phương pháp sử dụng đặc trưng thủ công như HOG, SIFT nên hạn chế đáng kể về khả năng khái quát khi điều kiện ánh sáng thay đổi hoặc hình ảnh bị nhiễu.

• Một số nghiên cứu gần đây như **ALPR-Net (2018)** đã chứng minh khả năng nhận dạng biển số xe với độ chính xác cao bằng Deep Learning. Mặc dù không tập trung vào xây dựng nền tảng web cho cộng đồng, đây là minh chứng rõ ràng cho tiềm năng của AI trong xử lý dữ liệu giao thông quy mô lớn.

Tổng kết từ các nghiên cứu trên, có thể thấy rằng mặc dù đã có nhiều hướng tiếp cận trong nhận dạng phương tiện, phát hiện chỗ trống và hỗ trợ tìm kiếm bãi đỗ xe, nhưng **chưa có hệ thống nào tích hợp đầy đủ các chức năng**, bao gồm:

- Nhận diện phương tiện bằng mô hình học sâu.
- Tự động phát hiện qua camera.
- Tra cứu thông tin phương tiện.

Chính vì vậy, đề tài này hướng đến việc phát triển một **nền tảng web**, kết hợp giữa nhận dạng phương tiện, quản lý bãi đỗ xe, tương tác người dùng và công nghệ thị giác máy tính hiện đại. Hệ thống không chỉ khắc phục những hạn chế của các nghiên cứu trước mà còn mở rộng khả năng ứng dụng vào thực tế, góp phần xây dựng môi trường đô thị thông minh và nâng cao trải nghiệm của người dùng trong việc tìm

kiểm và quản lý phương tiện.

3.Mục tiêu đề tài

Mục tiêu của đề tài là *xây dựng một ứng dụng web tích hợp trí tuệ nhân tạo*, cho phép người dùng quản lý bãi đỗ phương tiện và tra cứu thông tin một cách tự động. Hệ thống tập trung vào hai năng lực cốt lõi: *nhận dạng phương tiện gồm tên hãng và loại xe, từ đó có thể lấy thông tin về chiều cao, cân nặng...*

Các mục tiêu cụ thể gồm:

1. Nhận dạng hãng xe từ ảnh:

- Xây dựng mô hình nhận diện hãng xe với R-CNN và CNN so sánh với YOLOv8.
- Huấn luyện mô hình với dữ liệu ảnh đã xử lý (resize, tăng sáng, chuẩn hóa...) nhằm tăng độ chính xác và khả năng tổng quát hóa.

2. Nhận diện loại xe từ ảnh:

- Áp dụng craft để nhận diện vị trí chữ và sử dụng các OCR như paddle, transformer, easy để so sánh và nhận diện chữ.

3. Xây dựng giao diện web tích hợp mô hình AI:

- Cho phép người dùng tải ảnh lên và nhận kết quả hãng xe và loại xe.
- Kết hợp fuzzy matching để trả về thông tin chính xác cũng như các thông tin của phương tiện
- Giao diện đơn giản, dễ sử dụng, thân thiện với người không chuyên.

4. Quản lý phương tiện:

- Hệ thống thực hiện phân loại phương tiện dựa trên cân nặng.
- Quản lý xem vị trí phương tiện trong bãi và thông tin của phương tiện

Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các kỹ thuật học sâu (deep learning), thiết kế các mô hình CNN và R-CNN cơ bản và so sánh với mô hình chuyên về nhận diện như YOLO.
- So sánh các OCR và sử dụng fuzzy matching để trả về kết quả chính xác.
- Hệ thống quản lý cơ bản .

Phạm vi nghiên cứu:

- Dữ liệu đầu vào: Ảnh các đuôi xe của phương tiện (xe hơi). Dữ liệu được xử lý qua các bước tiền xử lý như resize, tăng sáng, chuẩn hóa, v.v.
- Nhận dạng logo hãng: Sử dụng R-CNN cơ bản để nhận diện vị trí logo hãng

Đề tài: Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính

sau đó sử dụng CNN cơ bản để phân loại logo sau đó so sánh với nhận diện của YOLO .

- Nhận diện tên hãng: Sử dụng Craft để nhận diện vị trí chữ nhận được biểu đồ
- nhiệt từ đó sử dụng các mô hình OCR để nhận diện chữ trong các vùng đã nhận diện và so sánh các OCR.
- Tìm kiếm thông tin bằng kết quả của OCR: OCR trả về kết quả tưởng đối nhò fuzzy matching có thể chuyển về kết quả đúng và lấy được thông tin phương tiện.

Giao diện người dùng: Website sử dụng HTML/CSS/JS.

4.Phương pháp nghiên cứu

Để thực hiện đề tài “Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính trong nhận dạng và quản lý phương tiện”, các phương pháp nghiên cứu chính được áp dụng bao gồm:

- **Phương pháp tổng hợp tài liệu lý thuyết:** Tìm hiểu và tổng hợp các kiến thức liên quan đến học sâu, CNN, R-CNN, YOLO, Craft, OCR, kỹ thuật xử lý ảnh, cũng như các tài liệu liên quan đến phương tiện (hãng phương tiện, loại phương tiện, thông tin phương tiện,...).
- **Phương pháp phân tích và khảo sát hệ thống thực tế:** Tham khảo các hệ thống bãi đỗ xe thông minh hiện có như Smart Parking, Parkopedia, hệ thống quản lý ra-vào bằng camera tại các trung tâm thương mại, nhằm đánh giá những chức năng đã tồn tại, hạn chế trong vận hành (như nhận dạng kém trong ánh sáng yếu, thiếu khả năng theo dõi thời gian thực), từ đó đề xuất hướng cải tiến phù hợp.
- **Phương pháp mô phỏng và lập trình thử nghiệm:** Áp dụng mô hình R-CNN, CNN, YOLOv8 để nhận dạng hãng của phương tiện; kết hợp OCR để trích xuất tên của phương tiện; sau đó sử dụng fuzzy matching để trả về kết quả chính xác cùng thông tin của phương tiện. Song song, mô phỏng sơ đồ bãi đỗ xe và hiển thị trạng thái chỗ trống – chỗ đã đỗ bằng giao diện web.
- **Phương pháp kiểm thử và đánh giá mô hình:** Tiến hành chia bộ dữ liệu thành tập huấn luyện và kiểm tra; đánh giá mô hình bằng các chỉ số như độ chính xác (accuracy), độ mất mát (loss), confusion matrix CER,...
- **Phương pháp tham khảo ý kiến chuyên gia và người dùng:** Tham khảo ý kiến giảng viên, kỹ sư trong lĩnh vực AI – camera giám sát và nhóm người dùng thử (tài xế, bảo vệ bãi xe) để hoàn thiện các tính năng như hiển thị chỗ đỗ, giao diện giám sát trực quan và quy trình quản lý nhằm giúp hệ thống dễ sử dụng, phù hợp thực tiễn và có tính ứng dụng cao.

5.Bố cục:

Phần I: Giới thiệu

Phần II: Nội dung

Chương 1. Mô tả bài toán: Trình bày tổng quan về vấn đề nghiên cứu, các nghiên cứu có liên quan, mục tiêu, phạm vi, phương pháp và hướng giải quyết bài toán đặt ra.

Chương 2. Thiết kế và cài đặt giải pháp: Trình bày chi tiết về kiến trúc hệ thống, các công nghệ được sử dụng, quy trình xử lý dữ liệu hình ảnh, thiết kế và huấn luyện mô hình học sâu, cùng với quá trình xây dựng website.

Chương 3. Đánh giá và giao diện: Tiến hành kiểm thử hệ thống với các tập dữ liệu thực tế, đánh giá độ chính xác và hiệu quả nhận dạng/tìm kiếm của mô hình.

Phần III: Phần kết luận

II. PHẦN NỘI DUNG

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

1.1. Mô tả chi tiết bài toán

Trong các hệ thống bãi đỗ xe hiện nay, đặc biệt là bãi đỗ xe nhiều tầng hoặc bãi đỗ xe tự động, việc phân bổ vị trí đỗ phù hợp với tải trọng của phương tiện đóng vai trò quan trọng trong việc đảm bảo an toàn kết cấu, tối ưu không gian sử dụng và kéo dài tuổi thọ công trình. Do đó, tại thời điểm phương tiện vào bãi, hệ thống thường yêu cầu xác định khối lượng (cân nặng) của xe để đưa ra quyết định sắp xếp vị trí đỗ tương ứng.

Giải pháp truyền thống phổ biến là sử dụng cân điện tử được lắp đặt tại cổng vào của bãi đỗ xe. Tuy nhiên, trong quá trình vận hành lâu dài, hệ thống cân này bộc lộ nhiều hạn chế. Cụ thể, các thiết bị cân thường xuyên chịu tác động của môi trường (bụi bẩn, độ ẩm, rung động cơ học) và tải trọng lớn, dẫn đến xuống cấp nhanh, sai lệch số đo và yêu cầu bảo trì, hiệu chuẩn định kỳ với chi phí cao. Khi cân hoạt động không chính xác, việc phân loại và bố trí xe có thể sai lệch, gây ảnh hưởng đến hiệu quả quản lý cũng như tiềm ẩn nguy cơ mất an toàn.

Trước những hạn chế nêu trên, bài toán đặt ra là cần một giải pháp thay thế cho hệ thống cân truyền thống, vừa đảm bảo độ tin cậy, vừa giảm chi phí vận hành và bảo trì trong dài hạn. Trong bối cảnh trí tuệ nhân tạo (Artificial Intelligence – AI) và thị giác máy tính (Computer Vision) đang phát triển mạnh mẽ, việc tận dụng camera giám sát kết hợp với các mô hình học sâu mở ra hướng tiếp cận mới cho bài toán này.

Cụ thể, hệ thống được đề xuất trong luận văn sử dụng camera để thu nhận hình ảnh phương tiện khi xe vào bãi, sau đó áp dụng các thuật toán AI để **nhận diện logo và hãng xe** (ví dụ: Toyota, Honda, Ford, Hyundai, v.v.). Dựa trên kết quả nhận diện này, hệ thống tiến hành tra cứu thông tin kỹ thuật của xe (bao gồm khối lượng bản thân của xe) từ cơ sở dữ liệu được xây dựng sẵn, thu thập từ các hãng sản xuất hoặc nguồn dữ liệu đáng tin cậy.

Qua cách tiếp cận này, khối lượng xe được ước lượng gián tiếp dựa trên đặc trưng hãng xe và dòng xe, thay vì đo trực tiếp bằng thiết bị cân vật lý. Kết quả ước lượng được xuất ra dưới dạng file dữ liệu (ví dụ: JSON, CSV hoặc cơ sở dữ liệu), phục vụ cho các module quản lý bãi đỗ xe như phân tầng, phân khu hoặc kiểm soát tải trọng.

1.1.1. Nhận dạng logo phương tiện từ hình ảnh:

- **Nhận dạng vị trí logo bằng R-CNN (Region-based Convolutional Neural Network):**
-

Đề tài: Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính

Trong bài toán phát hiện logo xe, hệ thống cần định vị (phát hiện vị trí) logo trong ảnh trước khi tiến hành phân loại. Ở bước này, mô hình R-CNN được sử dụng như một giải pháp thay thế cho các phương pháp truyền thống, giúp mô phỏng khả năng tìm kiếm mẫu dựa trên các phép tích chập có kernel được tối ưu hóa.

R-CNN có khả năng:

- + Xác định vùng có khả năng chứa logo trên thân xe.
- + Phát hiện các logo nhỏ, nghiêng, hoặc bị che khuất một phần.
- + Tạo ra bounding box bao quanh logo, giúp chuẩn hóa vùng ảnh đầu vào cho bước phân loại.
- + Ảnh đầu vào có thể là ảnh chụp xe tại bãi đỗ, camera giao thông hoặc hệ thống giám sát. Mục tiêu là tách đúng vùng chứa logo, ngay cả khi ánh sáng, góc chụp hoặc khoảng cách thay đổi.

➤ Phân loại logo xe bằng CNN:

Sau khi R-CNN phát hiện và cắt vùng chứa logo, mô hình CNN sẽ đảm nhiệm việc phân loại logo thành đúng hãng xe tương ứng, ví dụ như: Toyota, Honda, Mercedes-Benz, Ford, Hyundai, Kia, VinFast,...

CNN được chọn cho giai đoạn phân loại vì:

- + Có khả năng tự động học đặc trưng nổi bật của từng logo (đường nét, hình dạng, màu sắc).
- + Độ chính xác cao hơn các phương pháp truyền thống như K-NN hoặc SVM.
- + Khả năng chịu nhiễu tốt, phù hợp với logo bị biến dạng, mờ hoặc bị ảnh hưởng bởi môi trường.

➤ Nhận diện logo xe bằng YOLO:

Nhận diện bằng YOLO (You Only Look Once) không chỉ dừng lại ở phân loại logo mà còn phát hiện vị trí logo trong ảnh. YOLO thực hiện đồng thời phát hiện (object detection) và phân loại, phù hợp với các tình huống thực tế như camera giao thông, camera bãi đỗ xe hoặc hệ thống giám sát.

Ảnh đầu vào có thể chứa nhiều xe và nhiều logo khác nhau; YOLO sẽ trả về bounding box và nhãn logo của từng xe.

Ưu điểm của YOLO:

- + Nhận diện thời gian thực (real-time)
- + Tìm được logo nhỏ, bị che một phần
- + Hoạt động tốt khi bối cảnh phức tạp

1.1.2. Nhận dạng và trích xuất tên phương tiện từ hình ảnh:

- Phát hiện vùng văn bản bằng CRAFT (Character Region Awareness for Text Detection):

Trong bài toán nhận diện tên phương tiện, bước đầu tiên là phát hiện các vùng chứa văn bản trên xe như *tên hãng*, *tên dòng xe*, *ký hiệu phiên bản* (*Vios*, *Civic*, *Ranger*, *CX-5*, *Seltos*, ...).

CRAFT được sử dụng nhờ khả năng định vị chính xác từng ký tự và nhóm ký tự, giúp:

- + Phát hiện văn bản ở nhiều vị trí khác nhau trên thân xe (đuôi xe, hông xe, lưới tản nhiệt...).
- + Xử lý tốt các trường hợp văn bản có độ cong, nghiêng, hoặc kích thước nhỏ.
- + Tạo ra bounding box chính xác cho từng từ hoặc cụm từ, đảm bảo chất lượng đầu vào cho bước OCR.
- + CRAFT giúp nhận diện văn bản trong điều kiện thực tế như ánh sáng kém, hình ảnh bị rung, nền phức tạp hoặc logo – chữ bị bám bụi.

- Nhận dạng ký tự bằng các mô hình OCR khác nhau:

Sau khi CRAFT phát hiện và cắt vùng chứa văn bản, hệ thống tiến hành chuyển ảnh chữ sang văn bản (text recognition) bằng các mô hình OCR. Mỗi loại OCR mang một thế mạnh riêng, giúp tăng độ chính xác khi nhận dạng tên phương tiện.

- EasyOCR – Nhận dạng văn bản nhanh và linh hoạt

EasyOCR phù hợp cho việc nhận tên xe vì:

- Nhận dạng được nhiều font chữ và kiểu chữ trên mẫu xe.
- Hoạt động tốt với ảnh thực tế độ phân giải trung bình.
- Tốc độ nhanh, dễ triển khai trực tiếp trên hệ thống web.

EasyOCR thường cho kết quả tốt với các mẫu xe phổ biến có font chữ đơn giản như: Vios, Winner, Accent, Innova, Ranger, ...

- PaddleOCR – Độ chính xác cao và mạnh với văn bản khó

PaddleOCR được sử dụng như mô hình OCR mạnh hơn nhờ:

- Kiến trúc sâu, được tối ưu hóa cho văn bản nhỏ hoặc chữ mờ.
- Khả năng nhận dạng chữ nghiêng, chữ bạc màu hoặc bị che khuất.
- Mô hình đa ngôn ngữ, phù hợp với xe nhập khẩu nhiều nước.

PaddleOCR thường xử lý tốt các tên xe dạng ký hiệu hoặc kết hợp chữ + số như:

- Transformer OCR – Nhận dạng thông minh theo ngữ cảnh

Các mô hình OCR dựa trên Transformer (ViTSTR, TrOCR, PARSeq...) được dùng để tăng độ chính xác nhờ:

- Khả năng hiểu ngữ cảnh, đoán chữ bị thiếu hoặc mờ.
- Mô phỏng "đọc chữ" tương tự con người với self-attention.
- Đặc biệt hiệu quả cho chữ khó đọc hoặc bị biến dạng.

Ví dụ: Nếu ảnh mờ chỉ thấy “C__ic”, Transformer có thể suy luận ra “Civic”.

Mô hình này rất phù hợp để nhận dạng tên phương tiện cao cấp với font đặc biệt hoặc chữ nghệ thuật.

➤ Fuzzy Matching (so khớp mờ) qua cơ sở dữ liệu phương tiện:

Sau khi thu được văn bản từ ảnh thông qua mô hình OCR (EasyOCR, PaddleOCR hoặc Transformer OCR), hệ thống vẫn phải đối mặt với các lỗi nhận dạng phổ biến như nhầm lẫn ký tự ($O \leftrightarrow 0$, $I \leftrightarrow 1$, $S \leftrightarrow 5$), thiếu ký tự, thừa khoảng trắng hoặc sai chính tả do hình ảnh mờ, nghiêng hoặc nhiễu.

Để khắc phục các sai lầm này và xác định tên phương tiện chính xác nhất, hệ thống áp dụng hai cơ chế so khớp chuỗi mạnh mẽ trong Python: **FuzzyWuzzy** và **difflib.SequenceMatcher**.

1.1.3. Xây dựng giao diện web hiển thị kết quả nhận dạng:

Hệ thống web cho phép người dùng tải lên một hình ảnh của phương tiện, sau đó tự động xử lý để trả về tên xe, hãng xe, trọng lượng và tầng đậu phù hợp.

Toàn bộ quy trình diễn ra nhanh chóng, hoàn toàn tự động và tối ưu cho bối cảnh ứng dụng thực tế như bǎi đỗ xe thông minh, hệ thống quản lý phương tiện, hoặc camera giao thông.

➤ Nhận diện phương tiện từ ảnh:

Người dùng chỉ cần chọn hoặc tải lên một ảnh phương tiện.

Hệ thống thực hiện chuỗi xử lý:

✓ Phát hiện logo xe bằng YOLO

→ giúp xác định **hãng xe** chính xác.

✓ Trích xuất chữ (OCR)

→ đọc được các chữ như “CX5”, “Vios”, “CR-V”, “Fadil”.

✓ Fuzzy Matching với cơ sở dữ liệu

→ tìm ra **model đúng** ngay cả khi OCR nhận sai hoặc thiếu chữ.

➤ **Trả về thông tin xe:**

Sau khi xác định được model, hệ thống trả về đầy đủ:

- **Tên xe (Model)**
- **Hãng xe (Brand)**
- **Trọng lượng kerb weight**
- **Kích thước xe**
- **Năm sản xuất**
- **Tầng đỗ phương tiện trong hệ thống**

Tất cả hiển thị trực quan ngay trên website.

1.2. Vấn đề giải pháp liên quan đến bài toán

1.2.1. Công cụ lập trình – Visual Studio Code

Trong đề tài này, Visual Studio Code (VS Code) được sử dụng làm công cụ chính để phát triển toàn bộ hệ thống nhận diện phương tiện và trang web quản lý. VS Code là một trình soạn thảo mã nguồn miễn phí, đa nền tảng do Microsoft phát triển, nổi bật nhờ giao diện trực quan, nhẹ và khả năng mở rộng mạnh mẽ.

Phần mềm hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như **Python**, **JavaScript**, **HTML/CSS**, rất phù hợp với mô hình hệ thống gồm backend xử lý AI và frontend web của đề tài. VS Code cung cấp nhiều tính năng giúp tối ưu hóa quá trình phát triển như:

- **IntelliSense**: gợi ý mã nguồn thông minh, tăng tốc độ viết code.
- **Tích hợp Git**: giúp theo dõi phiên bản và quản lý mã nguồn thuận tiện.
- **Debugging**: hỗ trợ gỡ lỗi trực tiếp cho Python và JavaScript.
- **Extensions**: cho phép cài thêm các tiện ích như Python, Pylance, Live Server, Prettier... giúp lập trình nhanh và hiệu quả hơn.

Ngoài ra, VS Code còn cho phép tùy chỉnh giao diện, phím tắt, môi trường làm việc và đồng bộ hóa cấu hình giữa các thiết bị. Nhờ những ưu điểm này, Visual Studio Code hoàn toàn đáp ứng tốt nhu cầu lập trình, thử nghiệm mô hình YOLO, OCR, fuzzy matching cũng như xây dựng giao diện web của hệ thống trong đề tài.

1.2.2. Ngôn ngữ lập trình

Python là một ngôn ngữ lập trình bậc cao, thông dịch, đa mục đích, được Guido van Rossum phát triển và ra mắt lần đầu vào năm 1991, nổi bật với cú pháp rõ ràng,

dễ đọc và dễ học, giúp lập trình viên tập trung nhiều hơn vào giải quyết vấn đề thay vì bị phân tán bởi những chi tiết phức tạp của cú pháp.

Trong đề tài này, Python được lựa chọn làm ngôn ngữ lập trình chính để xây dựng hệ thống nhận dạng phương tiện nhờ vào tính đơn giản, mạnh mẽ và hệ sinh thái thư viện phong phú. Python là ngôn ngữ lập trình bậc cao, thông dịch, đa mục đích, nổi bật với cú pháp dễ đọc, giúp rút ngắn thời gian phát triển và tăng hiệu quả trong xử lý dữ liệu – hình ảnh.

Python đặc biệt phù hợp với các bài toán thị giác máy tính trong đề tài nhờ các điểm mạnh:

- **Thư viện AI – Machine Learning phong phú:** PyTorch, TensorFlow, scikit-learn hỗ trợ huấn luyện và triển khai mô hình CNN, R-CNN, OCR và YOLO.
- **Xử lý ảnh mạnh mẽ** với OpenCV, PIL, NumPy.
- **Thư viện OCR chuyên dụng** như EasyOCR, PaddleOCR, OCR Transformer.
- **Thư viện fuzzy matching** như FuzzyWuzzy, rapidfuzz, difflib để so khớp tên phương tiện sau OCR.
- **Khả năng tích hợp Web** thông qua Flask hoặc FastAPI, giúp đưa mô hình AI chạy trực tiếp trên backend của hệ thống bãi đỗ xe.

Python còn sở hữu cộng đồng lập trình rộng lớn, tài liệu phong phú và hàng trăm nghìn gói mở rộng trên PyPI, giúp việc xây dựng, thử nghiệm và triển khai hệ thống trở nên nhanh chóng và hiệu quả. Nhờ những ưu điểm trên, Python hoàn toàn đáp ứng tốt yêu cầu của một hệ thống nhận diện phương tiện tự động tích hợp OCR và fuzzy matching trong bãi đỗ xe thông minh.

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản tiêu chuẩn được sử dụng để xây dựng và tổ chức nội dung trên các trang web, đóng vai trò là “bộ khung” cơ bản mà mọi trang web đều dựa vào. Ra đời từ những năm đầu của World Wide Web, HTML được thiết kế để mô tả cấu trúc thông tin bằng các thẻ (tags), mỗi thẻ đảm nhiệm một chức năng cụ thể như tiêu đề, đoạn văn, hình ảnh, liên kết, bảng, biểu mẫu... HTML không phải là ngôn ngữ lập trình, mà là ngôn ngữ đánh dấu, cho phép trình duyệt hiểu và hiển thị nội dung một cách có tổ chức. Phiên bản hiện đại nhất, HTML5, đã mở rộng khả năng của HTML với nhiều tính năng mới như hỗ trợ đa phương tiện (audio, video) mà không cần plugin, các thẻ ngữ nghĩa (semantic tags) giúp cải thiện SEO, khả năng tương tác mạnh hơn thông qua API tích hợp và hỗ trợ ứng dụng web phức tạp. HTML thường được kết hợp với CSS để định dạng và làm đẹp giao diện, cùng JavaScript để thêm tính năng động, từ đó tạo nên các trang web hiện đại, trực quan và thân thiện với người dùng. Nhờ tính đơn giản, dễ học và vai trò nền tảng trong phát triển web, HTML là kỹ năng cơ bản mà bất kỳ lập trình viên hoặc nhà thiết kế web nào cũng cần nắm vững.

CSS (Cascading Style Sheets) là ngôn ngữ định kiểu được sử dụng để mô tả cách hiển thị và trình bày của các phần tử HTML trên trang web, đóng vai trò “trang điểm” cho bộ khung HTML. Ra đời vào giữa những năm 1990, CSS giúp tách biệt phần nội dung (HTML) và phần trình bày (style), cho phép lập trình viên và nhà thiết kế kiểm soát các yếu tố như màu sắc, kích thước, kiểu chữ, khoảng cách, bố cục, hiệu ứng và thậm chí là các hoạt ảnh phức tạp. CSS hoạt động theo nguyên tắc “cascading” – nghĩa là các quy tắc định kiểu có thể được kế thừa, ghi đè hoặc kết hợp, giúp việc quản lý giao diện trở nên linh hoạt. Phiên bản hiện đại, CSS3, đã bổ sung nhiều tính năng mạnh mẽ như bo góc, bóng đổ, gradient, hiệu ứng chuyển động, media queries (thiết kế web đáp ứng – responsive design), hỗ trợ biến CSS và nhiều module chuyên biệt khác. CSS có thể được viết trực tiếp trong thẻ HTML, đặt trong phần `<style>` của tài liệu, hoặc lưu thành file .css riêng để tái sử dụng và quản lý dễ dàng. Khi kết hợp với HTML và JavaScript, CSS tạo nên trải nghiệm web sinh động, bắt mắt và tương thích với nhiều thiết bị khác nhau, từ màn hình máy tính, máy tính bảng đến điện thoại thông minh.

JavaScript (thường viết tắt là JS) là một ngôn ngữ lập trình kịch bản đa năng, chủ yếu được sử dụng để tạo sự tương tác và tính năng động cho các trang web, biến những trang HTML và CSS tĩnh trở nên sống động và phản hồi theo hành vi của người dùng. Ra đời vào năm 1995 bởi Brendan Eich tại Netscape, JavaScript ban đầu chỉ hoạt động trên trình duyệt, nhưng với sự phát triển của công nghệ, đặc biệt là Node.js, nó đã trở thành một ngôn ngữ lập trình toàn diện có thể chạy cả phía máy khách (client-side) lẫn phía máy chủ (server-side). JavaScript hỗ trợ lập trình hướng đối tượng, lập trình hàm và sự kiện, đồng thời sở hữu kho thư viện và framework phong phú như React, Angular, Vue, Express... giúp tăng tốc quá trình phát triển ứng dụng. Các tính năng hiện đại của JavaScript như AJAX, Fetch API, WebSocket, hay ES6+ với cú pháp ngắn gọn hơn, đã giúp lập trình viên dễ dàng xây dựng ứng dụng web thời gian thực, trò chơi trực tuyến, ứng dụng di động và cả phần mềm desktop. Khi kết hợp với HTML và CSS, JavaScript trở thành một trong ba trụ cột của phát triển web, đóng vai trò then chốt trong việc mang lại trải nghiệm mượt mà, trực quan và tương tác cho người dùng.

1.2.3.Các giải pháp và thuật toán sử dụng trong hệ thống.

1.2.3.1.Giải pháp học sâu R-CNN (Region-based Convolutional Neural Network)

R-CNN là một trong những kiến trúc học sâu quan trọng trong bài toán phát hiện đối tượng, được thiết kế để xử lý hiệu quả các đối tượng nhỏ, dễ bị che khuất hoặc biến dạng – rất phù hợp trong bối cảnh camera bãi đỗ xe và giám sát giao thông. Điểm đặc trưng của R-CNN nằm ở cơ chế đề xuất vùng (Region Proposal) kết hợp với mạng CNN để trích xuất đặc trưng và phân loại, giúp mô hình tập trung chính xác vào

những khu vực có khả năng chứa logo xe.

Trong đề tài, R-CNN được sử dụng để **phát hiện và khoanh vùng vị trí logo** trên đuôi xe trong điều kiện thực tế như ánh sáng yếu, góc quay nghiêng, logo bị mờ hoặc dính bụi. Nhờ khả năng phân tích từng vùng ảnh độc lập, R-CNN có thể:

- Xác định chính xác logo ngay cả khi logo chỉ chiếm diện tích rất nhỏ trong khuôn hình.
- Chống chịu tốt với biến dạng, nghiêng hoặc che khuất một phần.
- Tạo ra bounding box chính xác để chuẩn hóa đầu vào cho bước phân loại hãng xe.

Cơ chế phát hiện theo vùng của R-CNN giúp mô hình duy trì độ ổn định và độ chính xác cao khi làm việc với dữ liệu camera giám sát – nơi điều kiện môi trường thay đổi liên tục và logo thường không rõ nét.

1.2.3.2. Giải pháp học sâu – CNN (Convolutional Neural Network)

CNN là một trong những mô hình cốt lõi của học sâu trong xử lý ảnh, hoạt động dựa trên các lớp tích chập nhằm trích xuất đặc trưng theo chiều sâu của ảnh. Trong hệ thống quản lý bãi đỗ xe thông minh, CNN được dùng làm mô hình phân loại logo và nhận diện hãng xe sau khi các vùng chứa logo được R-CNN hoặc YOLOv8 tách ra.

Lợi thế của CNN là quá trình huấn luyện nhanh, dễ triển khai và hiệu quả cao khi phân loại đối tượng có hình dạng tương đối cố định như logo xe. Mặc dù ít sâu hơn các kiến trúc hiện đại như ResNet hay ViT, CNN vẫn giữ được sự cân bằng giữa tốc độ và độ chính xác – yếu tố quan trọng trong hệ thống chạy thời gian thực.

YOLOv8 (You Only Look Once – phiên bản 8) là mô hình phát hiện đối tượng tiên tiến được phát triển bởi Ultralytics, nổi bật với tốc độ xử lý cực nhanh và độ chính xác cao, đặc biệt trong các ứng dụng giám sát thời gian thực.

Trong đề tài, YOLOv8 đảm nhận vai trò:

- Phát hiện xe trong khung hình camera bãi đỗ
- Xác định vị trí logo xe để đưa vào mô hình CNN/R-CNN phân loại
- Nhận diện nhiều đối tượng cùng lúc trong môi trường phức tạp
- Làm việc tốt khi đối tượng nhỏ, nghiêng, hoặc bị che khuất một phần

Với ưu điểm hiệu năng vượt trội, YOLOv8 là mô hình chủ lực giúp hệ thống hoạt động ổn định ở cả bãi đỗ xe tầng hầm lẫn môi trường ánh sáng ngoài trời.

1.2.3.3.1.Các chỉ số đánh giá mô hình: Precision, Recall, F1-score, CER:

- **Precision (Độ chính xác):** Precision đo lường tỷ lệ dự đoán dương tính (positive) đúng so với tổng số dự đoán dương tính mà mô hình đưa ra.

$$Precision = \frac{TP}{TP + FP}$$

Trong đó :

- TP (*True Positive*): Số mẫu dương tính dự đoán đúng.
- FP (*False Positive*): Số mẫu âm tính nhưng bị dự đoán nhầm là dương tính.

Ý nghĩa: Precision cao nghĩa là khi mô hình dự đoán một mẫu là dương tính, khả năng dự đoán đúng là cao.

- **Recall (Độ bao phủ/Độ nhạy):** Recall đo lường tỷ lệ dự đoán dương tính đúng so với tổng số mẫu thực tế là dương tính.

$$Recall = \frac{TP}{TP + FN}$$

Trong đó: FN (*False Negative*): Số mẫu dương tính nhưng bị dự đoán nhầm là âm tính.

Ý nghĩa: Recall cao nghĩa là mô hình phát hiện được hầu hết các mẫu dương tính.

- **F1-score:** F1-score là trung bình điều hòa (harmonic mean) của Precision và Recall, dùng để cân bằng giữa hai chỉ số này.

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Ý nghĩa: F1-score hữu ích khi dữ liệu mất cân bằng giữa hai lớp, và cần đánh giá cả khả năng chính xác lẫn bao phủ.

- **Chỉ số đánh giá CER (Character Error Rate):**

CER (Character Error Rate) là chỉ số dùng để đo độ chính xác của mô hình OCR bằng cách so sánh chuỗi ký tự dự đoán với chuỗi ký tự đúng.

Công thức CER:

$$CER = \frac{S+D+I}{N}$$

Trong đó:

- **S – Substitution:** số ký tự bị thay sai
- **D – Deletion:** số ký tự bị bỏ sót

- **I** – Insertion: số ký tự bị nhận nhầm thêm vào
- **N** – Tổng số ký tự của văn bản đúng

1.2.3.4. Tách vùng văn bản – CRAFT

CRAFT (Character Region Awareness for Text Detection) là mô hình phát hiện vùng chữ viết dựa trên nhận diện từng ký tự thay vì toàn bộ văn bản. Điều này giúp CRAFT xử lý hiệu quả các biển số hoặc văn bản trên thân xe bị cong, nghiêng hoặc méo hình.

Trong hệ thống, CRAFT được sử dụng để:

- Tách vùng biển số xe làm đầu vào cho OCR
- Phát hiện chữ ngay cả khi biển số bị bẩn, cũ hoặc ánh sáng phản chiếu
- Loại bỏ các vùng nhiễu, tăng độ chính xác nhận dạng chữ

CRAFT đóng vai trò quan trọng trong pipeline OCR, giúp các mô hình nhận dạng văn bản phía sau hoạt động ổn định hơn.

1.2.3.5. Giải pháp nhận diện chữ - OCR (Optical Character Recognition)

OCR (Optical Character Recognition – Nhận dạng ký tự quang học) là công nghệ giúp máy tính có thể tự động đọc và chuyển đổi văn bản từ hình ảnh hoặc video thành dạng văn bản số. Công nghệ này phân tích các vùng chứa ký tự, nhận diện các chữ cái, số và ký hiệu, sau đó chuyển chúng thành chuỗi văn bản để máy tính tiếp tục xử lý.

OCR được ứng dụng rộng rãi trong nhiều lĩnh vực như: đọc biển số xe, số hóa tài liệu, nhận dạng chữ viết tay, trích xuất thông tin từ hóa đơn, hay các hệ thống giám sát giao thông. Với sự phát triển của học sâu (Deep Learning), OCR hiện đại ngày càng chính xác hơn, đặc biệt khi kết hợp với mô hình trích xuất vùng chữ như **CRAFT**, và các bộ nhận dạng ký tự mạnh như **OCR Easy**, **OCR PaddleOCR**, hay **OCR Transformer**.

Trong hệ thống quản lý bãi đỗ xe thông minh, OCR là một thành phần quan trọng, đảm nhiệm vai trò nhận diện tên phương tiện từ hình ảnh chứa chữ sau khi trích xuất vùng chữ, các mô hình OCR tiến hành đọc và chuyển ký tự thành văn bản số.

Ba công nghệ OCR chính:

- **OCR Easy**: nhanh, gọn, phù hợp xử lý thời gian thực.
- **PaddleOCR**: hỗ trợ nhiều ngôn ngữ, độ chính xác cao, mạnh với ảnh nhiễu.
- **Transformer OCR**: sử dụng kiến trúc Attention → nhận dạng tốt các chuỗi ký tự dài, văn bản khó.

Nhờ OCR, hệ thống web có thể:

- ✓ Tự động suy ra **tên xe + hãng xe**
- ✓ Kết hợp fuzzy matching để tìm **thông tin xe, cân nặng, vị trí tầng đậu**
- ✓ Tự động hóa quản lý phương tiện khi ra/vào bãi

OCR giúp hệ thống trở nên thông minh, chính xác và tự động hơn, không cần nhập tay dữ liệu.

1.2.3.6. Công nghệ Fuzzy Matching

Fuzzy Matching là kỹ thuật so sánh chuỗi linh hoạt giúp tìm ra mức độ giống nhau giữa hai chuỗi văn bản **ngay cả khi chúng không hoàn toàn khớp nhau**.

Fuzzy matching được xây dựng dựa trên các thuật toán như:

- Levenshtein Distance (độ đo số phép sửa cần thiết)
- Ratio similarity (tỷ lệ ký tự trùng khớp)
- Token sort / token set (loại bỏ ảnh hưởng của thứ tự từ)

Nhờ đó, nó có khả năng đánh giá mức độ gần đúng giữa các chuỗi để tìm ra tên phù hợp nhất.

Trong hệ thống quản lý bãi đỗ xe thông minh, fuzzy matching là bước không thể thiếu khi kết hợp với OCR.

Vì OCR khi nhận dạng chữ trên xe có thể gặp vấn đề:

- Nhận sai 1–2 ký tự
- Bị nhiễu do bóng đổ, phản chiếu
- Chữ in nghiêng, cong
- Tên xe bị che khuất một phần

→ Nếu so khớp theo cách chính xác tuyệt đối thì hầu hết đều thất bại.

Fuzzy matching giải quyết các vấn đề này bằng cách:

- ✓ Khớp tên xe từ văn bản OCR với cơ sở dữ liệu chuẩn

Ví dụ OCR trả về:

- “TOYOYA VI0S” → khớp với “Toyota Vios”
- “CX 3” → khớp với “Mazda CX-3”

- ✓ Giúp hệ thống truy xuất thông tin liên quan

Bao gồm:

- Hàng xe
- Dòng xe
- Kích thước
- Cân nặng
- Chiều dài rộng cao
- Quy định tầng đậu

1.2.4. Các thư viện

PyTorch: PyTorch là một framework học sâu (deep learning framework) mã nguồn mở do Facebook AI Research (FAIR) phát triển với các chức năng chính: Xây dựng, huấn luyện và triển khai các mô hình học máy và học sâu. Cùng với các đặc điểm: **Tính toán động (Dynamic Computation Graph)**, đồ thị tính toán được tạo trong quá trình chạy, linh hoạt cho debug và thử nghiệm. Tích hợp mạnh với GPU qua CUDA. Hỗ trợ nhiều mô-đun sẵn có như torch.nn, torch.optim, torchvision cho xử lý ảnh. Ứng dụng của PyTorch: phân loại ảnh, xử lý ngôn ngữ tự nhiên, thị giác máy tính, học tăng cường.

OpenCV: OpenCV (*Open Source Computer Vision Library*) là thư viện mã nguồn mở chuyên về xử lý ảnh và thị giác máy tính dùng để đọc, ghi, và xử lý ảnh/video. Các phép biến đổi hình ảnh (resize, crop, lọc nhiễu...). Nhận dạng vật thể, khuôn mặt, trích xuất đặc trưng. **Đặc điểm nổi bật:** Hỗ trợ nhiều ngôn ngữ (Python, C++, Java...). Tối ưu cho hiệu năng thời gian thực. Ứng dụng: tiền xử lý ảnh trong bài toán nhận dạng chòm sao, ngoại hành tinh; trích xuất đặc trưng trước khi đưa vào mô hình học sâu.

NumPy: NumPy (*Numerical Python*) là thư viện Python cho tính toán khoa học và xử lý mảng số liệu. Chức năng chính là cung cấp cấu trúc dữ liệu ndarray (mảng đa chiều) tối ưu về tốc độ và bộ nhớ. Thư viện toán học phong phú: đại số tuyến tính, thống kê, phép biến đổi Fourier. **Đặc điểm nổi bật:** Hoạt động nhanh hơn danh sách Python thông thường nhờ được tối ưu bằng C. Tích hợp dễ dàng với các thư viện khác như OpenCV, PyTorch, Pandas. Ứng dụng: xử lý dữ liệu vector đặc trưng, tính toán ma trận, chuẩn hóa dữ liệu đầu vào.

Scikit-learn: scikit-learn là thư viện Python mã nguồn mở dành cho học máy(machine learning). Chức năng chính: Thuật toán học có giám sát (classification, regression). Thuật toán học không giám sát (clustering, dimensionality reduction). Tiện ích tiền xử lý (chuẩn hóa, mã hóa dữ liệu). Các công cụ đánh giá mô hình (precision, recall, F1-score, confusion matrix...). **Đặc điểm nổi bật:** API đơn giản, dễ sử dụng. Hỗ trợ nhiều mô hình kinh điển (SVM, KNN, Decision Tree...). Ứng

Đề tài: Xây dựng hệ thống bài đồ xe thông minh ứng dụng thị giác máy tính

dụng: đánh giá mô hình nhận dạng chòm sao, tính toán các chỉ số hiệu suất, chia dữ liệu train/val/test.

TensorFlow là một framework học sâu (deep learning framework) mã nguồn mở được phát triển bởi Google Brain Team, ra mắt lần đầu vào năm 2015. TensorFlow cung cấp một hệ sinh thái toàn diện phục vụ xây dựng, huấn luyện và triển khai các mô hình học máy và học sâu trên nhiều nền tảng khác nhau như CPU, GPU, TPU và cả thiết bị di động. Tương tự PyTorch, TensorFlow hỗ trợ định nghĩa mô hình bằng cách sử dụng các toán tử đại số trên tensor, đồng thời tích hợp cơ chế tự động đạo hàm (automatic differentiation) giúp tối ưu hóa mô hình một cách hiệu quả.

TensorFlow sở hữu API cấp cao **Keras**, cho phép xây dựng mô hình nhanh, dễ sử dụng và trực quan, đồng thời vẫn hỗ trợ API mức thấp để thực hiện các thao tác tùy chỉnh chuyên sâu. Một trong những ưu điểm nổi bật của TensorFlow là khả năng triển khai mô hình linh hoạt thông qua TensorFlow Serving, TensorFlow Lite và TensorFlow.js, giúp các ứng dụng AI dễ dàng tích hợp vào môi trường sản xuất (production).

Ultralytics là thư viện mã nguồn mở do Ultralytics phát triển, cung cấp các mô hình YOLO (You Only Look Once) hiện đại như YOLOv5, YOLOv8 và YOLOv11, phục vụ cho các bài toán thị giác máy tính như phát hiện đối tượng (object detection), phân đoạn ảnh (segmentation), nhận dạng tư thế (pose estimation) và theo dõi đối tượng (tracking).

người dùng có thể dễ dàng khởi tạo, huấn luyện và suy luận các mô hình YOLO một cách trực quan và hiệu quả mà không cần cấu hình phức tạp.

Thư viện được xây dựng tối ưu để hoạt động tốt trên nhiều nền tảng, bao gồm CPU, GPU (NVIDIA CUDA), Google Colab, và có thể triển khai trên server, thiết bị edge hoặc các hệ thống nhúng. Ultralytics cung cấp mô hình YOLOv8 — thế hệ mới nhất — với nhiều cải tiến như tốc độ xử lý nhanh, độ chính xác cao, tối ưu hóa kiến trúc backbone và head, cũng như hỗ trợ multi-task đồng thời.

IPython.display – Image, display: Trong môi trường lập trình tương tác như Jupyter Notebook hoặc Google Colab, thư viện IPython.display cung cấp các công cụ trực quan để hiển thị nội dung đa phương tiện, bao gồm hình ảnh, âm thanh, video, HTML và nhiều định dạng khác.

người dùng có thể dễ dàng hiển thị trực tiếp hình ảnh ngay trong notebook mà không cần sử dụng các thư viện vẽ phức tạp. Lớp Image cho phép tải ảnh từ đường dẫn, URL hoặc dữ liệu nhị phân, trong khi hàm display() giúp trình bày ảnh trong ô notebook một cách trực quan và rõ ràng.

Matplotlib – matplotlib.pyplot as plt: là một module thuộc thư viện Matplotlib – một trong những thư viện trực quan hóa dữ liệu phổ biến và mạnh mẽ nhất trong Python.

Người dùng có thể tạo ra hàng loạt biểu đồ khác nhau như biểu đồ đường (line chart), biểu đồ cột (bar chart), biểu đồ phân tán (scatter plot), biểu đồ ảnh (imshow),... Thư viện được thiết kế theo phong cách tương tự MATLAB, giúp việc trực quan hóa dữ liệu trở nên dễ dàng và linh hoạt.

PaddleOCR là một thư viện OCR (Optical Character Recognition) mã nguồn mở do Baidu phát triển dựa trên framework PaddlePaddle. Thư viện được thiết kế để cung cấp một hệ thống OCR hoàn chỉnh “end-to-end”, bao gồm cả **phát hiện văn bản (text detection)** và **nhận dạng văn bản (text recognition)** với độ chính xác cao và tốc độ nhanh.

EasyOCR là một thư viện nhận dạng ký tự quang học (OCR – Optical Character Recognition) mã nguồn mở do nhóm Jaided AI phát triển. Đây là một giải pháp OCR “nhẹ”, dễ sử dụng và không yêu cầu cấu hình phức tạp, phù hợp cho các ứng dụng thực tế cần tốc độ xử lý nhanh và triển khai đơn giản.

TrOCR (Transformer-based Optical Character Recognition) là một mô hình OCR tiên tiến được phát triển bởi Microsoft và tích hợp trong thư viện **Hugging Face Transformers**. Không giống các mô hình OCR truyền thống dựa trên CNN + RNN, TrOCR sử dụng kiến trúc **Vision Transformer (ViT)** kết hợp với **Transformer Decoder**, tạo thành mô hình **VisionEncoderDecoderModel** – chuyên dùng cho xử lý ảnh thành text.

Difflib là một thư viện chuẩn (standard library) của Python được thiết kế nhằm hỗ trợ việc **so sánh chuỗi (string comparison)** và **tìm mức độ tương đồng** giữa các đoạn văn bản. Thư viện này được giới thiệu từ những phiên bản đầu của Python và thường được sử dụng trong các bài toán xử lý văn bản, phân tích sự thay đổi giữa các file, hoặc trong các hệ thống cần xác định các chuỗi tương tự nhau.

difflib hoạt động dựa trên các thuật toán so khớp chuỗi tối ưu (đặc biệt là thuật toán Ratcliff/Obershelp – tìm subsequence chung), từ đó đánh giá mức độ giống nhau giữa hai đoạn text theo dạng **tỉ lệ tương đồng (similarity ratio)**. Ngoài ra, thư viện còn hỗ trợ tìm kiếm chuỗi giống nhất trong một danh sách, tạo báo cáo so sánh chi tiết (diff), hoặc gợi ý chỉnh sửa tương tự như hệ thống kiểm tra lỗi chính tả.

FuzzyWuzzy là một thư viện mã nguồn mở trong Python được phát triển nhằm giải quyết các bài toán **so khớp chuỗi mờ (fuzzy matching)**, đặc biệt hiệu quả trong những trường hợp văn bản chứa lỗi chính tả, ký tự bị thiếu, dư hoặc bị OCR nhận sai. Thư viện này hoạt động dựa trên **Levenshtein Distance** – một thuật toán đo khoảng cách chỉnh sửa giữa hai chuỗi bằng cách tính số lượng phép thay thế, thêm hoặc xoá ký tự cần thiết để biến chuỗi A thành chuỗi B.

Ra mắt từ năm 2011, FuzzyWuzzy nổi bật nhờ cú pháp đơn giản, dễ sử dụng, nhưng đạt hiệu quả cao trong nhiều tình huống thực tế như chuẩn hoá dữ liệu, tìm kiếm tên gần đúng, ghép dữ liệu khách hàng, hoặc đối chiếu văn bản trong hệ thống nhận dạng.

Pandas là một thư viện mã nguồn mở trong Python được thiết kế dành cho **xử lý, phân tích và quản lý dữ liệu dạng bảng**. Ra mắt lần đầu vào năm 2008 bởi Wes McKinney, Pandas nhanh chóng trở thành công cụ tiêu chuẩn trong cộng đồng khoa học dữ liệu nhờ khả năng làm việc hiệu quả với các tập dữ liệu lớn, thao tác linh hoạt và tốc độ xử lý cao.

1.3.Cách đánh giá mô hình

Để đảm bảo mô hình nhận dạng và hoạt động hiệu quả, cần có những tiêu chí và phương pháp đánh giá rõ ràng. Trong phạm vi đề tài này, mô hình được đánh giá dựa trên các chỉ số chuẩn trong lĩnh vực học sâu và xử lý ảnh:

- **Độ chính xác (Accuracy):** đo lường tỷ lệ dự đoán đúng trên tổng số mẫu kiểm tra. Đây là thước đo cơ bản phản ánh khả năng phân loại tổng quát của mô hình.

Độ chính xác theo lớp (Precision): cho biết trong số các mẫu mà mô hình dự đoán là thuộc một lớp nào đó, có bao nhiêu mẫu thực sự đúng. Chỉ số này quan trọng trong việc hạn chế các dự đoán sai dương tính (false positive).

- **Độ bao phủ (Recall):** cho biết trong số các mẫu thực sự thuộc về một lớp, mô hình có phát hiện được bao nhiêu phần trăm. Recall quan trọng trong việc đảm bảo mô hình không bỏ sót các đối tượng cần nhận dạng.

- **F1-Score:** là trung bình điều hòa giữa Precision và Recall, giúp cân bằng hai chỉ số này khi mô hình có sự chênh lệch.

- **Hàm mất mát (Loss Function – CrossEntropyLoss):** theo dõi mức độ sai lệch giữa dự đoán của mô hình và nhãn thực tế trong quá trình huấn luyện. Giá trị loss càng nhỏ cho thấy mô hình học càng tốt.

- **Biểu đồ huấn luyện (Training/Validation Curves):** theo dõi sự thay đổi của Accuracy và Loss trên cả tập huấn luyện và tập kiểm thử (validation) qua từng epoch, giúp phát hiện hiện tượng overfitting hoặc underfitting.

- **CER (Character Error Rate):** là chỉ số dùng để đo độ chính xác của mô hình OCR bằng cách so sánh chuỗi ký tự dự đoán với chuỗi ký tự đúng.

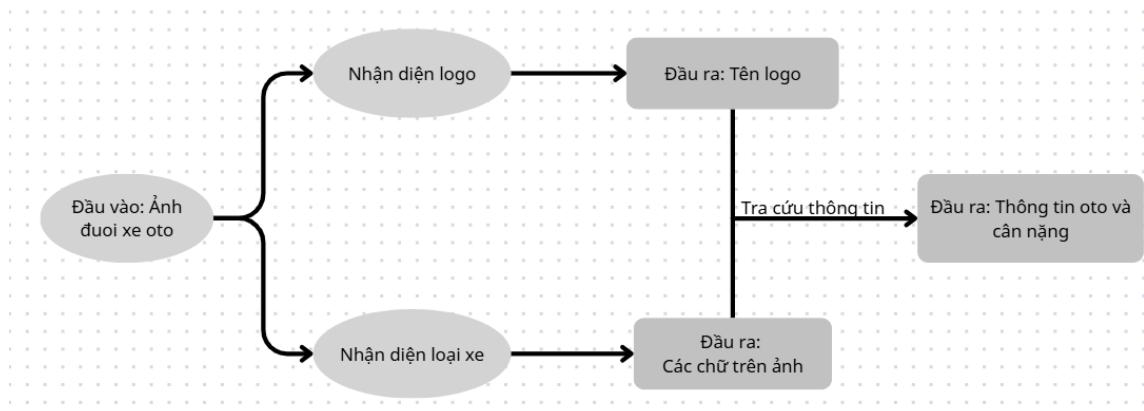
CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

2.1.Thiết kế hệ thống

Hệ thống đề xuất sử dụng ảnh đuôi xe thu nhận từ camera làm đầu vào để tự động xác định thông tin phương tiện. Từ ảnh đầu vào, hệ thống thực hiện nhận diện logo hãng xe, qua đó xác định tên hãng sản xuất. Đồng thời, hệ thống tiến hành nhận diện chữ và ký hiệu dòng xe xuất hiện trên đuôi xe bằng các thuật toán thị giác máy tính và OCR.

Dựa trên hai thông tin thu được là hãng xe và loại/dòng xe, hệ thống tiến hành tra cứu cơ sở dữ liệu kỹ thuật được xây dựng sẵn để suy ra các thông tin chi tiết của phương tiện, bao gồm đặc biệt là khối lượng xe. Kết quả cuối cùng được xuất ra dưới dạng dữ liệu có cấu trúc, phục vụ cho bài toán quản lý và phân loại xe trong bãi đỗ.

2.1.1.Sơ đồ tổng quát



Hình 2.1: Sơ đồ tổng quát của hệ thống

Sơ đồ mô tả một hệ thống tra cứu thông tin xe hơi gồm bốn thành phần chính, hoạt động phối hợp để phân tích ảnh đuôi xe và trả kết quả chi tiết cho người dùng.

Thành phần đầu tiên là khói xử lý ảnh đầu vào, trong đó hệ thống tiếp nhận ảnh đuôi xe ô tô và bắt đầu tách thành hai luồng xử lý: một luồng nhận diện logo và một luồng nhận diện loại xe, chữ hoặc ký hiệu xuất hiện trên xe.

Thành phần thứ hai là các mô hình Deep Learning, gồm nhiều mô-đun riêng biệt:

- R-CNN dùng để xác định vị trí logo trên ảnh.
- CNN dùng để phân loại chính xác loại logo → từ đó xác định hãng xe.
- YOLO đảm nhiệm nhận diện vị trí và phân loại loại xe.
- Craft và các bộ OCR (EasyOCR, PaddleOCR hoặc Transformer OCR) dùng để nhận diện chữ xuất hiện trên đuôi xe (model, phiên bản, mã ký hiệu,...). Các mô hình này hoạt động độc lập nhưng có thể kết hợp đầu ra với nhau để cung cấp thông tin đầy đủ nhất.

Thành phần thứ ba là *khối xử lý logic trung tâm*, nơi tiếp nhận toàn bộ đầu ra từ các mô hình deep learning. Tại đây hệ thống thực hiện tổng hợp thông tin, so sánh và đối chiếu bằng thuật toán fuzzy matching với tập dữ liệu có sẵn nhằm xác định chính xác tên xe, loại xe, kích thước (dài–rộng–cao), năm sản xuất và cân nặng. Khối này cũng chịu trách nhiệm chuẩn hóa dữ liệu, loại bỏ nhiễu và chọn ra kết quả phù hợp nhất.

Thành phần cuối cùng là *tập dữ liệu mô tả xe hơi* (file data hoặc cơ sở dữ liệu), nơi lưu trữ thông tin về các hãng xe, model, kích thước, thông số kỹ thuật và dữ liệu nhận diện hỗ trợ. Thành phần này được truy vấn bởi khối logic khi cần tra cứu, đồng thời có thể được cập nhật thêm khi mô hình nhận diện thu được dữ liệu mới.

Bốn thành phần trên phối hợp nhịp nhàng: ảnh đuôi xe được đưa vào, mô hình deep learning phân tích, khối logic đổi chiều và hoàn thiện thông tin, sau đó kết quả cuối cùng gồm tên hãng xe, model, kích thước và các dữ liệu kỹ thuật được xuất ra cho người dùng hoặc cho hệ thống khác sử dụng.

2.1.2. Giao diện người dùng

Frontend là lớp giao tiếp trực tiếp với người dùng, được xây dựng nhằm mang lại trải nghiệm thao tác đơn giản, trực quan và nhanh chóng, đáp ứng yêu cầu quản lý bãi đỗ xe theo thời gian thực. Giao diện được thiết kế theo phong cách UI/UX hiện đại với tông màu xanh, tím nhẹ nhàng, tạo cảm giác thân thiện nhưng vẫn đảm bảo tính chuyên nghiệp của một hệ thống giám sát.

Các thành phần chính trên frontend bao gồm:

► Khu vực tải ảnh (Upload Section):

Cho phép người dùng kéo – thả hoặc chọn ảnh từ thiết bị để hệ thống thực hiện nhận dạng xe.

Vùng upload được thiết kế dạng khung nét đứt, có biểu tượng minh họa và hướng dẫn định dạng ảnh (JPG, PNG, JPEG). Ảnh sau khi tải lên sẽ được hiển thị ngay bên dưới để người dùng kiểm tra trước khi tiến hành xử lý.

► Nút thực hiện nhận diện (Action Button):

Nút “Thực hiện nhận” giúp người dùng khởi động quá trình nhận diện biển số, loại xe và gợi ý vị trí đỗ phù hợp. Nút được làm nổi bật bằng màu gradient và bố trí ở vị trí dễ bấm, hỗ trợ thao tác nhanh trong môi trường vận hành thực tế.

► Bảng trạng thái bãi đỗ (Parking Status Dashboard):

Hiển thị tổng quan tình trạng toàn bộ bãi đỗ xe theo từng tầng:

- Tầng 1: xe dưới 1000kg

- Tầng 2: xe từ 1000–2000kg
- Tầng 3: xe trên 2000kg

Mỗi thẻ trạng thái cho biết số xe đã đậu, số vị trí còn trống, tổng số chỗ và phần trăm sử dụng. Các thẻ được phối màu khác nhau để người dùng dễ dàng phân biệt và theo dõi nhanh.

➤ **Danh sách vị trí đỗ xe (Slot Viewer):**

Dưới khu vực trạng thái là bảng liệt kê chi tiết các ô đỗ theo từng tầng. Mỗi ô hiển thị mã vị trí (ví dụ: I.A, I.B, I.C...) kèm trạng thái *Empty* hoặc *Occupied*. Giao diện hỗ trợ cuộn dọc và chuyển đổi tab giữa các tầng, đảm bảo hiển thị rõ ràng ngay cả khi số lượng vị trí lớn.

➤ **Thanh trạng thái hệ thống (System Status Bar):**

Nằm ở góc trên bên phải, cung cấp thông tin vận hành theo thời gian thực:

- Trạng thái kết nối (“Hệ thống trực tuyến”)
- Tổng số xe đang đậu / sức chứa tối đa

Đây là khu vực quan trọng giúp quản trị viên nhanh chóng đánh giá tình trạng chung của bãi đỗ.

➤ **Thông tin phương tiện:**

Gồm các trường như:

- + Hãng xe (Trước và sau khi chuẩn hóa)
- + Tên xe (Trước và sau khi chuẩn hóa)

Dữ liệu được tự động điền dựa trên quá trình nhận dạng logo, tên xe, text OCR và thuật toán chuẩn hóa.

Việc hiển thị cả “trước” và “sau” giúp người vận hành đối chiếu, kiểm tra độ chính xác của kết quả

➤ **Thông tin cân nặng và vị trí đậu:**

Bao gồm:

- Khối lượng ước tính (dựa trên model hoặc dữ liệu chuẩn)
- Tầng phù hợp theo phân loại trọng lượng
- Mã vị trí đậu được đề xuất
- ID xe được gán bởi hệ thống

Đề tài: Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính

Bảng được chia hai cột đối xứng giúp dễ quan sát và phân nhóm thông tin hợp lý.

➤ Tác vụ xác nhận:

Hai nút chức năng chính được đặt nổi bật phía dưới:

- Xác nhận đã đỗ lại – dành cho trường hợp xe vào bãi và vị trí được ghi nhận mới.
- Phương tiện rời đi – cập nhật trạng thái slot về Empty và ghi log rời bãi.

Các nút sử dụng màu sắc đặc trưng (xanh và đỏ) để giảm nhầm lẫn khi thao tác nhanh.

➤ Danh sách các xe đã đỗ (Parked Vehicles List):

Khu vực này hiển thị danh sách thời gian thực các phương tiện hiện đang đậu trong bãi, được trình bày dạng danh sách cuộn (scrollable list).

Mỗi mục bao gồm:

- Tên xe
- ID phương tiện
- Vị trí đậu (ở bên phải)

Giao diện đơn giản nhưng rõ ràng, giúp quản trị viên tra cứu nhanh mà không cần chuyển trang. Danh sách tự động cập nhật sau mỗi lần thêm/xóa xe.

➤ Hệ thống quản lý (System Management Panel):

Khu vực này cung cấp các chức năng phục vụ vận hành và giám sát.

Nhóm tác vụ điều khiển

Gồm ba nút chính:

- Xóa lịch sử – xóa toàn bộ log để làm sạch dữ liệu khi cần.
- Khởi động lại tất cả – reset trạng thái toàn bộ bãi đỗ.
- Xem thông tin phương tiện – truy cập trang chi tiết hoặc popup hiển thị lịch sử.

Các nút dùng màu sắc phân loại (cam, đỏ, xám) để thể hiện mức độ ưu tiên và cảnh báo.

➤ Tải về dữ liệu:

Cho phép export lịch sử đỗ xe dưới dạng CSV/Excel để phục vụ mục đích thống kê hoặc báo cáo.

➤ Bảng thống kê tổng hợp:

Hiển thị các chỉ số quan trọng của hệ thống:

- Tổng số lượt xe
- Số xe hiện tại
- Sức chứa tối đa (capacity)
- Số lượt xe trong ngày

Phần thống kê được tách thành các ô có viền trắng, nền sáng, bố trí dạng lưới giúp việc đọc số liệu dễ dàng và trực quan.

2.2.Cài đặt giải pháp

2.2.1.Thu thập dữ liệu

Bộ dữ liệu được xây dựng nhằm phục vụ cho bài toán nhận diện hằng xe dựa trên hình ảnh phần đuôi xe. Dữ liệu được thu thập hoàn toàn thủ công từ các nguồn đáng tin cậy, bao gồm:

- Trang chủ chính thức của từng hằng xe
- Các trang tin tức ô tô (VnExpress Auto, AutoPro, CarAndDriver,...)
- Các trang thương mại điện tử ô tô (oto.com.vn, bonbanh.com,...)
- Hình ảnh chất lượng cao từ Google Image, có chọn lọc để đảm bảo đúng đuôi xe và đúng mẫu thật

Tổng cộng có 10 hằng xe, mỗi hằng thu thập 50 ảnh, tổng cộng 500 ảnh. Tất cả ảnh đều được kiểm tra thủ công để bảo đảm:

- Chỉ chứa đuôi xe (rear view)
- Không lẫn ảnh đầu xe, ảnh nội thất, ảnh bị che khuất quá nhiều
- Hình ảnh đủ rõ để mô hình học đặc trưng .
- Không trùng lặp

Name
1-Vinfast
2-Toyota
3-Hyundai
4-Ford
5-Mitsubishi
6-Mazda
7-Kia
8-Honda
9-Mercedes
10-Suzuki

Hình 2.2: Dữ liệu thu thập

2.2.2. Tiền xử lý dữ liệu

Trước khi sử dụng dữ liệu ảnh cho các tác vụ nhận dạng xe, phân loại hãng xe hoặc phát hiện vị trí đỗ xe dựa trên mô hình R-CNN, CNN và YOLOv8, cần thực hiện bước tiền xử lý nhằm chuẩn hóa và nâng cao chất lượng dữ liệu đầu vào. Các bước tiền xử lý cụ thể như sau:

➤ Chuyển đổi định dạng ảnh:

Tất cả ảnh thu thập từ camera bãi đỗ xe và các nguồn dữ liệu bổ sung đều được chuyển về định dạng phổ biến như .jpg hoặc .png.

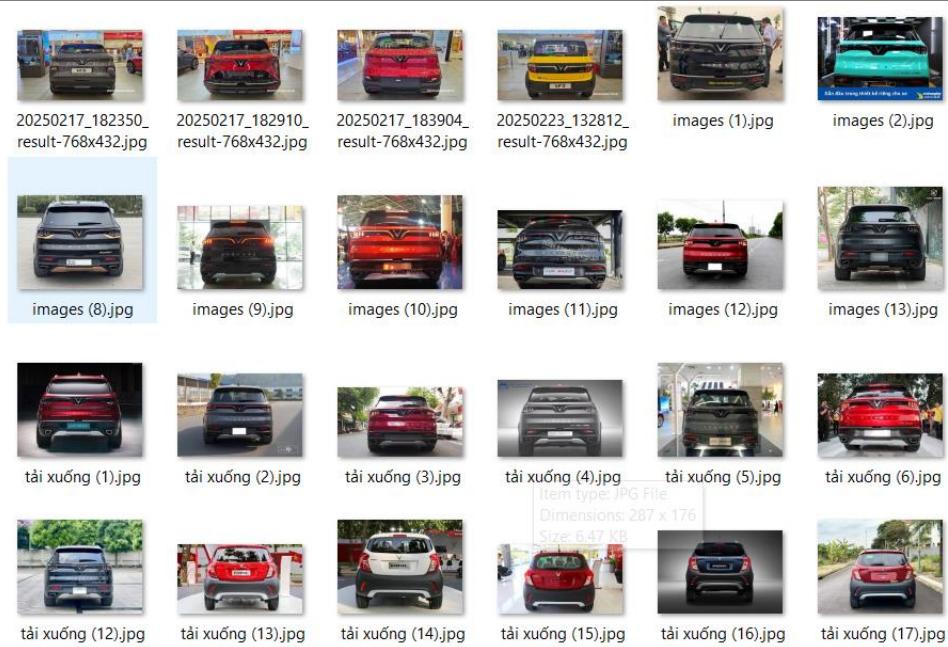
Điều này đảm bảo:

+Tương thích với các thư viện xử lý ảnh như OpenCV, Pillow, PyTorch.

+Giảm dung lượng lưu trữ so với các định dạng gốc (BMP, TIFF).

+Dễ dàng tích hợp vào pipeline huấn luyện và suy luận.

Đề tài: Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính



Hình 2.3: Dữ liệu đã qua kiểm tra

➤ Chuẩn hóa kích thước ảnh:

Ảnh được resize về 224×224 pixel, phù hợp với các CNN tiêu chuẩn.

Việc cố định kích thước giúp:

+Đồng bộ đầu vào cho mô hình,

+Giảm số chiều đặc trưng,

+Tăng tốc độ huấn luyện.

Đối với YOLOv8 để phát hiện đuôi xe / biển số / vị trí đỗ xe

Ảnh được resize về **640×640 pixel** (chuẩn của YOLO), đảm bảo:

- Giữ tỷ lệ phù hợp để bounding box không bị biến dạng,
- Cân bằng giữa độ chi tiết và tốc độ dự đoán,
 - Tối ưu hóa khả năng phát hiện nhiều xe trong cùng khung hình.

➤ Gán nhãn dữ liệu:

Dữ liệu thu thập từ bãi đỗ xe được gán nhãn theo mục đích huấn luyện:

- **Nhận dạng hãng xe:** mỗi ảnh được gán nhãn một trong 10 hãng như VinFast, Toyota, Hyundai,...
- **Phát hiện đối tượng (YOLOv8):** mỗi ảnh có file nhãn dạng .txt chứa vị trí bounding box của **logo hãng xe**, theo chuẩn YOLO.

```
dataset/
└── train/
    ├── images/
    |   ├── img1.jpg
    |   ├── img2.jpg
    |   └── ...
    └── labels/
        ├── img1.txt
        ├── img2.txt
        └── ...
└── val/
    ├── images/
    └── labels/
└── test/
    └── images/
```

Hình 2.4: Cấu trúc dữ liệu để huấn luyện R-CNN

➤ **Tăng cường dữ liệu (Data Augmentation):**

Để mô hình có khả năng tổng quát hóa tốt trong điều kiện camera thực tế (ánh sáng thay đổi, mưa, rung máy, góc lệch,...), tập dữ liệu được tăng cường bằng các phép biến đổi: xoay ảnh ngẫu nhiên $\pm 15^\circ$, lật ngang, thay đổi độ sáng – tương phản, nhiễu Gaussian, phóng to / thu nhỏ



Hình 2.5: Ảnh sau khi tăng cường

2.2.3. Huấn luyện mô hình nhận diện logo:

2.2.3.1. Quy trình huấn luyện mô hình R-CNN:

Mô hình R-CNN (Region-based Convolutional Neural Network) được huấn luyện theo ba giai đoạn chính: Đề xuất vùng (Region Proposal), Trích xuất đặc trưng (Feature Extraction) và Phân loại + hồi quy hộp bao (Classification & Bounding Box Regression). Quy trình này giúp mô hình phát hiện chính xác vị trí logo xe trong ảnh và tạo ra bounding box đúng nhất trước khi chuyển sang bước phân loại.

➤Region Proposal (Đề xuất Vùng):

Trong bước này, hệ thống tạo ra các vùng ứng viên (candidate regions) nhằm xác định những khu vực có khả năng chứa logo. Các bước chính gồm:

- Chia ảnh gốc thành nhiều vùng kích thước khác nhau dựa trên chiều dài và chiều rộng ảnh.

Các kích thước cửa sổ trượt thường dùng: 32×32 , 64×64 , 96×96 , 128×128 , ...

Điều này giúp mô hình có thể phát hiện logo dù kích thước lớn nhỏ khác nhau.

- Sinh ngẫu nhiên 50–100 vùng ứng viên (proposal regions) trên toàn bộ ảnh.

Các vùng này bao phủ nhiều vị trí và nhiều tỷ lệ khung hình khác nhau.

- Loại bỏ các vùng trùng nhau hoặc không hợp lệ (ví dụ: quá nhỏ, méo, hoặc tỉ lệ không phù hợp).

- Quá trình này giúp giảm số lượng vùng cần đưa vào CNN.

Đề tài: Xây dựng hệ thống bái đỡ xe thông minh ứng dụng thị giác máy tính

Kết quả của bước này là tập hợp các ROI (Region of Interest) có thể chứa logo để đưa vào bước trích xuất đặc trưng.

►Feature Extraction (Trích xuất Đặc trưng):

Sau khi có các vùng ROI, mỗi vùng sẽ được xử lý độc lập bằng một mạng CNN dùng để trích xuất vector đặc trưng.

Các bước chi tiết:

- Duyệt qua từng vùng đề xuất (ROI) và kiểm tra tính hợp lệ (không quá nhỏ, không trống, tỷ lệ phù hợp).
- Chuẩn hóa kích thước vùng ROI về một kích thước cố định (ví dụ: 224×224 px) để phù hợp với đầu vào của CNN.

Trích xuất đặc trưng bằng CNN cơ bản, gồm các tầng:

- Conv2D
- MaxPooling2D
- Conv2D
- MaxPooling2D
- Conv2D
- GlobalAveragePooling2D
- Dense

Các tầng này giúp trích xuất các đặc trưng quan trọng của logo: đường cong, cạnh, góc, kết cấu, độ tương phản,...

Tạo ra vector đặc trưng cho mỗi vùng ROI.

Đây là đầu vào cho bộ phân loại và bộ hồi quy bounding box ở bước tiếp theo.

►Classification & Bounding Box Regression (Phân loại & Hồi quy hộp bao):

Mỗi vector đặc trưng trích xuất từ ROI sẽ được đưa vào hai nhánh huấn luyện riêng biệt:

(a) Classification — Phân loại vùng ROI

- Mục tiêu: Xác định xem vùng đó có phải là logo hay không phải logo.
- Sử dụng bộ phân loại softmax hoặc sigmoid tùy theo số lớp.
- Kết quả: vùng được gán nhãn “logo” hoặc bị loại bỏ.

(b) Bounding Box Regression — Hồi quy hộp bao

Đề tài: Xây dựng hệ thống bái đỡ xe thông minh ứng dụng thị giác máy tính

- Mục tiêu: Điều chỉnh (refine) tọa độ bounding box sao cho khớp chính xác với vị trí logo.
- Mạng hồi quy dự đoán 4 giá trị:
x_center, y_center, width, height
- Bounding box cuối cùng được tinh chỉnh dựa trên special loss function (Smooth L1 Loss).

Kết quả của bước này là:

- Vị trí logo chính xác (bounding box refined)
- Xác định ROI nào thật sự chứa logo

Quá trình huấn luyện:

```
[2/642] Processing: 00e81eacd79891.jpg.rf.a00474d78ad4f4d3a0be4ded3eea9916.jpg
Image shape: (626, 908, 3)
Ground truth: [431.9, 280.2, 53.6, 32.8]
Generating region proposals...
Generated 400 regions
Extracting features...
    Extracting features: 0/400 regions
    Extracting features: 50/400 regions
    Extracting features: 100/400 regions
    Extracting features: 150/400 regions
    Extracting features: 200/400 regions
    Extracting features: 250/400 regions
    Extracting features: 300/400 regions
    Extracting features: 350/400 regions
Creating training samples...
Added 0 positive, 400 negative samples
```

Hình 2.6: Quá trình huấn luyện R-CNN lần 1

Kết luận: mô hình hoàn toàn không học được gì với R-CNN cơ bản.

➤Chỉnh sửa thông số R-CNN lần thứ 1:

Cải tiến đề xuất vùng: chia nhỏ hơn thành 16, 24, 32, 48, 64, 80, 96, 128

Tăng số lượng vùng từ 400 thành 2000 .

Giảm điều kiện positive từ $\text{IoU} > 0.5 \rightarrow 0.3$

Tăng negative sample threshold từ $\text{IoU} < 0.2 \rightarrow < 0.1$

Quá trình huấn luyện:

```
Extracting features: 0/2000 regions
Extracting features: 900/2000 regions
Extracting features: 950/2000 regions
Extracting features: 1000/2000 regions
Extracting features: 1050/2000 regions
Extracting features: 1100/2000 regions
Extracting features: 1150/2000 regions
Extracting features: 1200/2000 regions
Extracting features: 1250/2000 regions
Extracting features: 1300/2000 regions
Extracting features: 1350/2000 regions
Extracting features: 1400/2000 regions
Extracting features: 1450/2000 regions
Extracting features: 1500/2000 regions
Extracting features: 1550/2000 regions
Extracting features: 1600/2000 regions
Extracting features: 1650/2000 regions
Extracting features: 1700/2000 regions
Extracting features: 1750/2000 regions
Extracting features: 1800/2000 regions
Extracting features: 1850/2000 regions
Extracting features: 1900/2000 regions
Extracting features: 1950/2000 regions
Creating training samples...
Added 0 positive, 0 negative samples
```

Hình 2.7: Quá trình huấn luyện R-CNN lần 2

Kết luận: vẫn không trích xuất được đặc trưng

➤Chỉnh sửa thông số R-CNN lần thứ 2:

Sử dụng Selective Search (thuật toán tìm kiếm có chọn lọc thay vì random như ban đầu) để tạo 200–1500 vùng đề xuất nhưng tập trung vào **edges** (viền sắc nét, những nơi thay đổi màu sắc, độ sáng), do đó xác xuất bao phủ/trùm lên logo cao hơn nhiều.

Giảm điều kiện positive IoU ≥ 0.3

Tăng negative IoU ≤ 0.05

Lấy **5–20 sample positive** (vùng chứa vật thể) bằng cách jitter ground truth (di chuyển nhẹ: tịnh tiến, xoay, scale) thêm $\pm 5–20$ pixel

Lấy **50–100 negative** (vùng không chứa vật thể) gần logo (IoU giữa 0.05–0.3)

Thêm các negative xa hơn → để mô hình học background

```
Extracting: 1450/2000
Extracting: 1500/2000
Extracting: 1550/2000
Extracting: 1600/2000
Extracting: 1650/2000
Extracting: 1700/2000
Extracting: 1750/2000
Extracting: 1800/2000
Extracting: 1850/2000
Extracting: 1900/2000
Extracting: 1950/2000
Added: 7 pos / 1984 neg
```

```
[5/642] Image: 033b7c76409a13.jpg.rf.0895699abcd76aa48093cd8a33c2d906.jpg
GT: [401.4000000000003, 500.4, 42.6000000000002, 46.7799999999997]
Regions: 2000
Extracting: 0/2000
Extracting: 50/2000
```

Hình 2.8: Quá trình huấn luyện R-CNN lần 2

=> Kết luận: mô hình đã học được đặc trưng tuy nhiên quá trình huấn luyện quá mệt thời gian do mô hình chạy quá chậm (khoảng 1 phút/ 1 ảnh).

➤Chỉnh sửa thông số R-CNN lần thứ 3:

Cải tiến mô hình CNN : cụ thể là mô hình ResNet18

ResNet-18 (Residual Network với 18 layers) là một kiến trúc mạng neural convolutional (CNN) được giới thiệu bởi Microsoft Research năm 2015. Nó giải quyết vấn đề vanishing gradient trong các mạng neural sâu.

Sử dụng kiến trúc Fast R-CNN tăng tốc độ xử lý

Chỉnh lại cấu trúc dataset cho phù hợp mới mô hình theo định dạng: image.jpg x_min y_min x_max y_max với image.jpg là tên file và x_min y_min x_max y_max là tọa độ bounding box logo hãng

```
99%|#####8| 633/642 [2:04:32<00:31, 3.48s/it]Loss: 6.3900  
99%|#####8| 634/642 [2:04:35<00:26, 3.33s/it]Loss: 4.9290  
99%|#####8| 635/642 [2:04:38<00:22, 3.27s/it]Loss: 4.5315  
99%|#####9| 636/642 [2:04:41<00:19, 3.19s/it]Loss: 8.2640  
99%|#####9| 637/642 [2:04:44<00:15, 3.09s/it]Loss: 7.5211  
99%|#####9| 638/642 [2:04:47<00:12, 3.01s/it]Loss: 5.9398  
100%|#####9| 639/642 [2:04:50<00:08, 2.96s/it]Loss: 7.6345  
100%|#####9| 640/642 [2:04:53<00:06, 3.05s/it]Loss: 8.1562  
100%|#####9| 641/642 [2:04:56<00:03, 3.07s/it]Loss: 7.1732
```

Hình 2.9: Quá trình huấn luyện R-CNN lần 4

Kết luận: Mô hình đã cải tiến đáng kể tuy nhiên vẫn còn nhiều sai sót và tốc độ cũng chưa được nhanh.

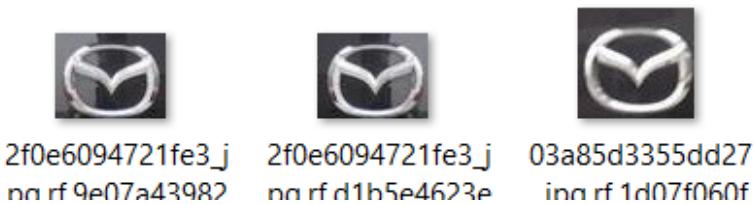
2.2.3.2. Quy trình huấn luyện mô hình CNN:

Mô hình CNN (Convolutional Neural Network) hoạt động dựa trên chuỗi các lớp tích chập (Convolution), kích hoạt (Activation) và gộp (Pooling) nhằm trích xuất các đặc trưng quan trọng trong ảnh. Quá trình học gồm hai giai đoạn chính: (1) Trích xuất đặc trưng – các lớp tích chập liên tục học những đường nét, kết cấu và hình dạng logo xe; (2) Phân loại – các đặc trưng được đưa vào các lớp kết nối đầy đủ (Fully Connected) để dự đoán nhãn logo.

Nhờ cơ chế trích xuất đặc trưng phân cấp này, CNN có thể phân biệt hiệu quả giữa các loại logo xe mặc dù có độ tương đồng cao, nhưng hạn chế ở chỗ chỉ phân loại, không xác định được vị trí của logo trong ảnh.

➤Xử lý dataset:

Thu thập dữ liệu logo và tăng cường: quay trái, quay phải phóng to thu nhỏ, làm mờ, ..



Hình 2.10: Dữ liệu huấn luyện CNN

➤**Thiết kế CNN:**

Trích xuất đặc trưng bằng CNN cơ bản, gồm các tầng:

Trích xuất đặc trưng bằng:

- Conv2D
- Dropout
- MaxPooling2D

Lặp lại 3 lần.

- Flatten

Ngăn chặn overfitting bằng :

- Dense
- Dropout

Lặp lại 2 lần và kết thúc bằng Dense.

Kết quả:

```
Epoch 2/30
21/21 ━━━━━━━━━━━━━━━━ 42s 2s/step - accuracy: 0.1175 - loss: 2.3022 - val_accuracy: 0.1132 - val_loss: 2.3007
Epoch 3/30
21/21 ━━━━━━━━━━━━━━━━ 43s 2s/step - accuracy: 0.1283 - loss: 2.3042 - val_accuracy: 0.1132 - val_loss: 2.3003
Epoch 4/30
21/21 ━━━━━━━━━━━━━━━━ 42s 2s/step - accuracy: 0.1049 - loss: 2.3069 - val_accuracy: 0.1132 - val_loss: 2.3010
Epoch 5/30
21/21 ━━━━━━━━━━━━━━━━ 42s 2s/step - accuracy: 0.1049 - loss: 2.3099 - val_accuracy: 0.1132 - val_loss: 2.3015
Epoch 6/30
21/21 ━━━━━━━━━━━━━━━━ 42s 2s/step - accuracy: 0.1418 - loss: 2.2983 - val_accuracy: 0.1132 - val_loss: 2.3017
```

Hình 2.11: Quá trình huấn luyện CNN lần 1

Kết luận: mô hình gần như không thể học có thể do việc ngăn chặn overfitting mà sử dụng dropout quá mức cần thiết.

➤**Chỉnh sửa CNN lần thứ 1:**

Thiết kế mô hình đơn giản hơn gồm các tầng:

Trích xuất đặc trưng bằng:

Đề tài: Xây dựng hệ thống bãi đỗ xe thông minh ứng dụng thị giác máy tính

o Conv2D

o MaxPooling2D

Lặp lại 2 lần. (loại bỏ dropout)

o Flatten

Ngăn chặn overfitting bằng :

o Densen

o Dropout

Kết thúc bằng Densen.

Kết quả:

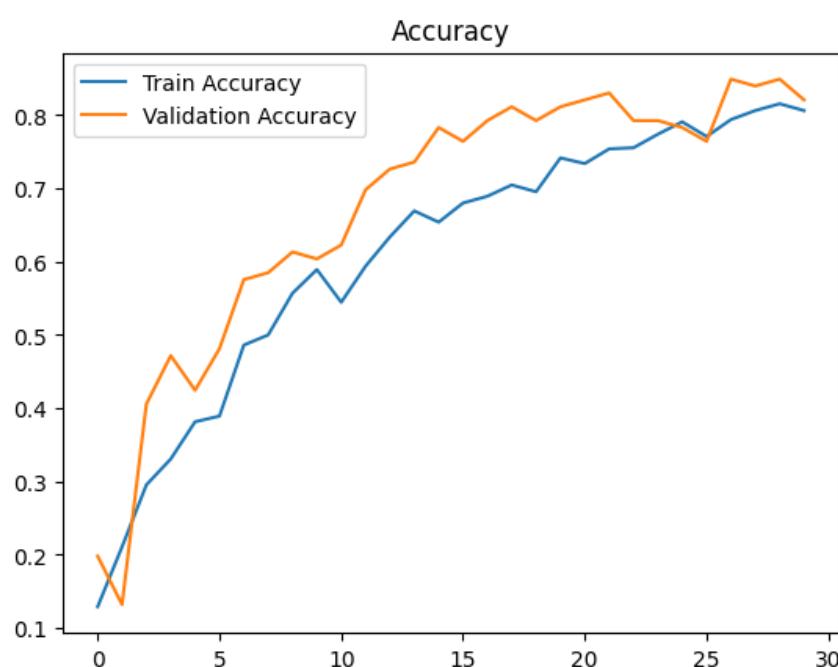
```
loss: 0.6470 - accuracy: 0.7938 - val_loss: 0.5871 - val_accuracy: 0.8491  
loss: 0.6245 - accuracy: 0.8062 - val_loss: 0.5492 - val_accuracy: 0.8396  
loss: 0.6275 - accuracy: 0.8154 - val_loss: 0.5272 - val_accuracy: 0.8491  
loss: 0.6301 - accuracy: 0.8062 - val_loss: 0.5088 - val_accuracy: 0.8208
```

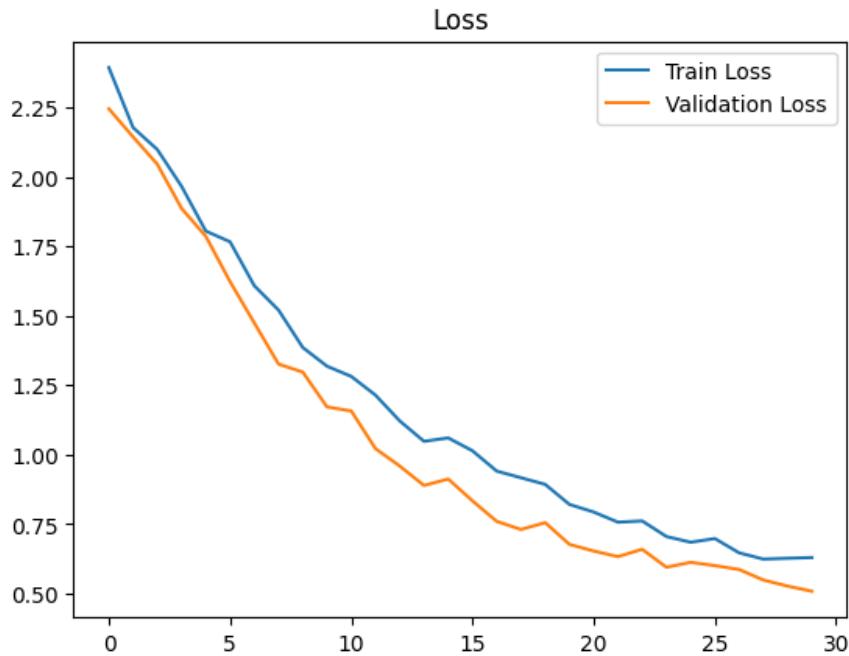
Hình 2.12: Quá trình huấn luyện CNN lần 2

Kết luận: mô hình nhận diện tương đối tốt và có thể sử dụng

➤Kiểm thử mô hình CNN:

Kiểm thử với tập test khoảng 50 ảnh.





Hình 2.13: Kết quả của CNN

➤Kết hợp R-CNN và CNN để nhận diện:

Mô hình R-CNN còn nhiều hạn chế chưa nhận diện được rõ vị trí logo dẫn đến dù mô hình CNN có cao đi nữa thì chung quy tổng thể vẫn không đạt yêu cầu

2.2.3.3.Quy trình huấn luyện mô hình YOLO:

YOLO (You Only Look Once) là mô hình phát hiện đối tượng thời gian thực hoạt động theo cơ chế “một lần nhìn toàn ảnh”. YOLO không tách riêng bước đề xuất vùng mà chia toàn bộ ảnh thành nhiều ô lưới (grid cells). Mỗi ô trực tiếp dự đoán:

- xác suất có đối tượng,
- tọa độ bounding box,
- nhãn lớp tương ứng.

Nhờ thực hiện cả phát hiện và phân loại trong một lần suy luận duy nhất, YOLO cho tốc độ xử lý rất nhanh (real-time) và vẫn duy trì độ chính xác cao. Điều này giúp mô hình nhận diện và khoanh vùng logo xe trên ảnh một cách liền mạch mà không cần trải qua các bước trung gian như R-CNN.

➤Huấn luyện mô hình YOLOv8 lần thứ 1:

Số epochs (số lần duyệt qua toàn bộ dữ liệu): 30

ImgSz (kích thước ảnh): 320

Batch (batch size): 8

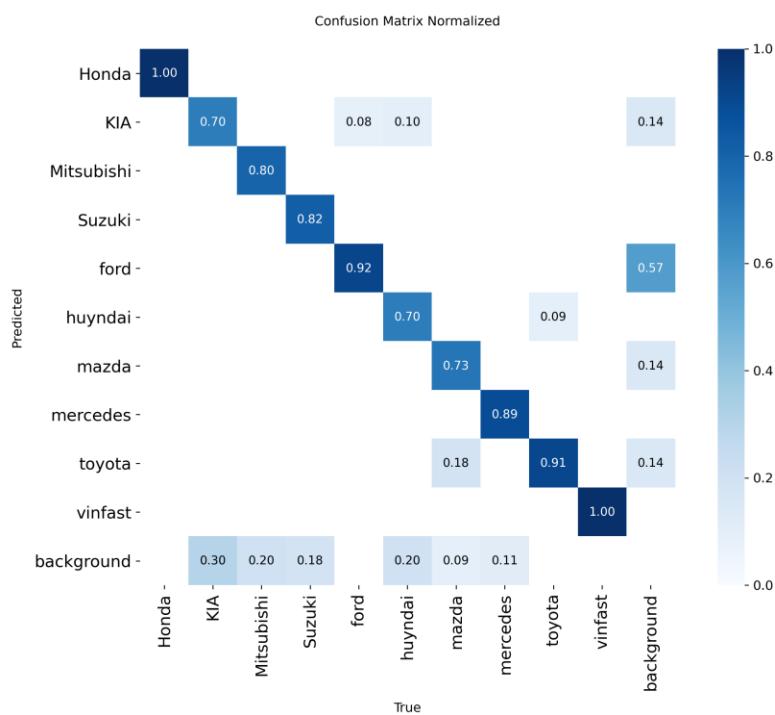
Đề tài: Xây dựng hệ thống bái đồ xe thông minh ứng dụng thị giác máy tính

Patien (kiên nhẫn):5

Lr0 (tốc độ học): 0.05

Ta thu được model tốt nhất: best.pt

Kết quả kiểm thử model với tập test khoảng 50 ảnh :



Hình 2.14: Kết quả lần 1 huấn luyện của YOLO

➤Huấn luyện mô hình YOLOv8 lần thứ 2:

Tăng số epochs để tăng số lần huấn luyện (cập nhập trọng số) của mô hình

Số epochs (số lần duyệt qua toàn bộ dữ liệu): 50

Imgsz (kích thước ảnh): 320

Batch (batch size): 8

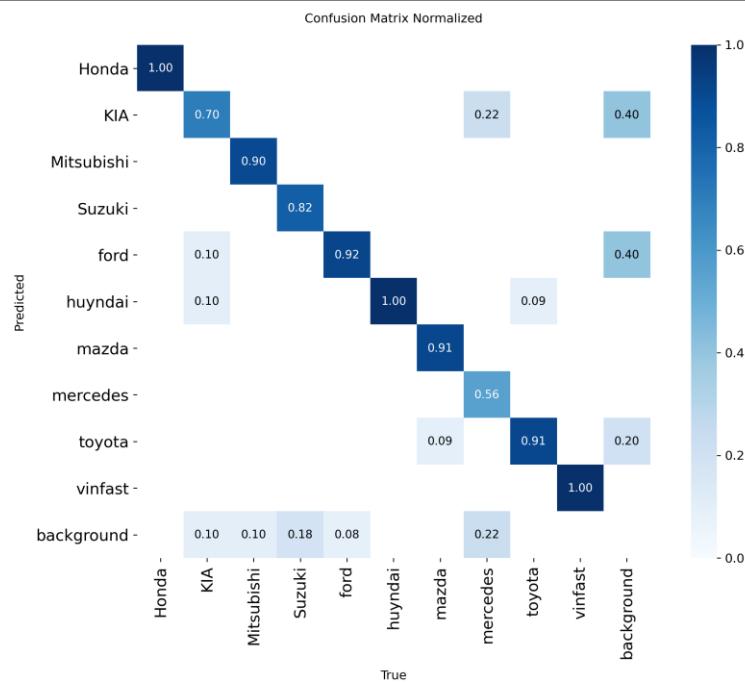
Patien (kiên nhẫn):10

Lr0 (tốc độ học): 0.05

Ta thu được model tốt nhất: best1.pt

Kết quả kiểm thử model best1.pt với tập test gồm 50 ảnh:

Đề tài: Xây dựng hệ thống bài đồ xe thông minh ứng dụng thị giác máy tính



Hình 2.15: Kết quả lần 2 huấn luyện của YOLO

Kết luận: mô hình YOLO sau hai lần train có kết quả tương tự nhau tuy nhiên lần có số epochs cao hơi thì có độ chính xác cao hơi.

➤So sánh nhận diện logo hãng xe của YOLO và K-CNN + CNN:

- Từ kết quả trả về thông qua quá trình huấn luyện ta dễ dàng nhận thấy YOLO tốt hơn hoàn toàn so với K-CNN + CNN.

	all	106	106	0.89	0.954	0.966	0.606
Honda	11	11	0.928	1	0.995	0.563	
KIA	10	10	0.747	0.886	0.906	0.566	
Mitsubishi	10	10	0.914	0.9	0.986	0.607	
Suzuki	11	11	0.99	1	0.995	0.577	
ford	12	12	0.773	1	0.936	0.575	
hyundai	10	10	0.794	1	0.978	0.618	
mazda	11	11	0.973	1	0.995	0.707	
mercedes	9	9	1	0.775	0.893	0.555	
toyota	11	11	0.844	0.983	0.981	0.646	
vinfast	11	11	0.936	1	0.995	0.646	

Speed: 0.2ms preprocess, 31.9ms inference, 0.0ms loss, 0.4ms postprocess per image
Results saved to /kaggle/working/runs/detect/val3

Validation completed!

mAP50: 0.9660

mAP50-95: 0.6060

Precision: 0.8899

Recall: 0.9544

Hình 2.16: Kết quả huấn luyện của YOLO

Kết luận: Nhờ vào các chỉ số vượt trội như là: Precision (độ chính xác), Recall (độ bao phủ), mAP50 (mean Average Precision tại IoU = 0.5, tức là độ chính xác trung

bình của các lớp khi độ trùng khớp giữa khung giữ đoán và thực tế là 0.5), mAP50-90(Trung bình mAP tại các ngưỡng IoU từ 0.5 đến 0.95).

=>Chọn mô hình YOLO lần huấn luyện thứ 2 làm mô hình nhận diện vị trí và tên logo của phương tiện

2.2.4. Nhận diện tên xe:

2.2.4.1 Craft (Character-Region Awareness for Text detection):

Hệ thống sử dụng mô hình CRAFT để xây dựng **công cụ nhận diện chữ và tạo heatmap** trực tiếp trên ảnh. CRAFT là một mô hình deep learning mạnh mẽ dựa trên CNN, cho phép phát hiện vùng chữ (text regions) với độ chính xác cao, ngay cả trên các ảnh phức tạp hoặc văn bản uốn cong, mà không cần bước tiền xử lý phức tạp.

>Hệ thống sử dụng framework VGG16 để xây dựng bộ trích xuất đặc trưng đa tầng:

-VGG16 là một kiến trúc mạng nơ-ron tích chập tiên tiến dựa trên PyTorch, cho phép trích xuất đặc trưng hình ảnh ở nhiều mức độ trừu tượng khác nhau mà không cần tiền xử lý phức tạp.

-Trong hệ thống, dữ liệu đặc trưng hình ảnh được xử lý qua 16 lớp tích chập sắp xếp thành 5 khối. Mỗi lớp tích chập trích xuất thông tin với mức độ phức tạp khác nhau: các lớp đầu tiên (conv1_1, conv1_2) phát hiện biên cạnh và gradient cơ bản; các lớp trung gian (conv3_1 đến conv3_3) nhận diện hình dạng và kết cấu; các lớp sâu (conv5_1 đến conv5_3) nắm bắt ngữ nghĩa và ngữ cảnh tổng thể. Dữ liệu đặc trưng được truyền qua mạng thông qua các phép toán tích chập (convolution) và chuẩn hóa theo batch (batch normalization). Mỗi đặc trưng được biểu diễn dưới dạng các bản đồ kích hoạt (activation maps) trong không gian nhiều chiều. Hệ thống pooling (max-pooling) giữa các khối giúp giảm kích thước không gian đồng thời tăng vùng tiếp nhận, cho phép mô hình nhận diện mẫu hình ở các tỷ lệ khác nhau.

>Hệ thống sử dụng checkpoint craft_mlt_25k.pth để nạp tri thức tiền huấn luyện:

-Craft_mlt_25k.pth là tệp trọng số mô hình đã được huấn luyện trên tập dữ liệu đa ngôn ngữ MLT-25k, chứa đựng tri thức về đặc trưng văn bản từ hàng nghìn mẫu huấn luyện.

-Các thành phần tri thức trong checkpoint gồm:

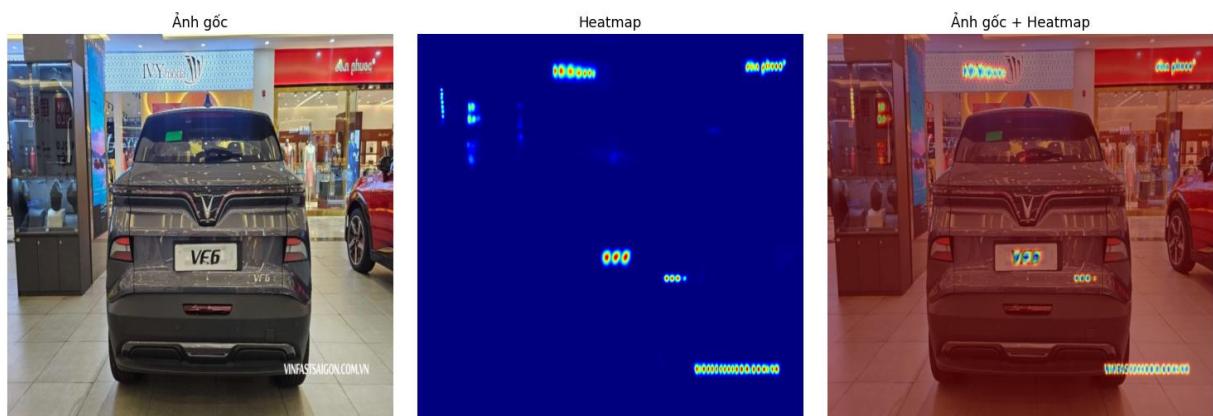
+Bộ Phân Loại Đặc Trung Văn Bản

+Mô Hình Quan Hệ Không Gian:

+Bộ Tích Hợp Đa Ngôn Ngữ:

➤Quá trình nhận diện vị trí của chữ:

-CRAFT (Character Region Awareness for Text Detection) là mô hình tạo bản đồ nhiệt hai chiều biểu thị xác suất xuất hiện của vùng ký tự và mối quan hệ giữa chúng.



Hình 2.17: Kết quả nhận diện của CRAFT

2.2.4.2 OCR (Optical Character Recognition) :

-**OCR (Optical Character Recognition)** là hệ thống công nghệ chuyển đổi văn bản từ định dạng hình ảnh (ảnh chụp, scan) thành định dạng văn bản số có thể chỉnh sửa, tìm kiếm và xử lý tự động.

➤Quá trình nhận diện chữ:

-Khởi tạo hệ thống:

Tải mô hình CRAFT với trọng số đã huấn luyện

Khởi tạo OCR với hỗ trợ ngôn ngữ Tiếng Anh

Thiết lập các tham số xử lý ảnh

-Tiền xử lý ảnh:

Chuyển đổi không gian màu: RGB → Grayscale

Tăng cường độ tương phản: Sử dụng CLAHE (Contrast Limited Adaptive Histogram Equalization)

Làm sắc nét: Áp dụng kernel convolution

Tái tạo màu: Grayscale → RGB

-Phát hiện văn bản:

Đầu vào: Ảnh đã tiền xử lý

Quá trình:

- Mô hình CRAFT phân tích ảnh tạo hai bản đồ:
 - Region Score Map: Xác suất pixel thuộc trung tâm ký tự
 - Affinity Score Map: Mối quan hệ giữa các ký tự
- Trích xuất bounding boxes đa giác bao quanh vùng văn bản

Đầu ra: Danh sách các vùng văn bản với tọa độ đa giác.

-*Tối ưu hóa vùng phát hiện chữ:*

Tính toán margin động: Dựa trên kích thước vùng phát hiện

Công thức: Margin = 33% kích thước vùng (tối thiểu 15 pixel)

Mục đích: Bao gồm toàn bộ ký tự và khoảng trống xung quanh

-*Nhận dạng văn bản:*

Crop vùng văn bản: Cắt phần ảnh tương ứng với bounding box

Nâng cao chất lượng ảnh crop:

Upscaling: Phóng to ảnh nhỏ (2-4 lần) bằng interpolation

Tăng cường độ sáng: Sử dụng CLAHE trên kênh L của LAB color space

Làm sắc nét: Áp dụng sharpening kernel

Khử nhiễu: Median filtering

-OCR với EasyOCR:

Truyền ảnh đã xử lý trực tiếp dưới dạng numpy array

Lấy kết quả có confidence cao nhất

Trích xuất văn bản và độ tin cậy

-OCR với PaddleOCR:

Lưu trữ tạm thời: Lưu vùng đã xử lý thành file ảnh

OCR processing:

- Truyền file ảnh vào PaddleOCR engine
- Kích hoạt text recognition pipeline

Xử lý kết quả:

- Trích xuất text và confidence score
- Xử lý multiple detection nếu có

Đánh giá độ tin cậy kết quả

-OCR với Transformer OCR:

Chuyển đổi từ mảng NumPy sang ảnh PIL (yêu cầu bắt buộc của TrOCR)

Tiền Xử Lý Cho Transformer

- Chuẩn hóa giá trị pixel: Điều chỉnh giá trị pixel theo đặc tả kỹ thuật của TrOCR
- Chuyển đổi sang tensor: Biến đổi dữ liệu ảnh sang định dạng tensor PyTorch
- Thêm chiều batch: Bổ sung chiều batch (batch dimension) cho đầu vào mô hình

Suy Luận Với Transformer

- Bộ mã hóa hình ảnh (Vision Encoder): Vision Transformer (ViT) trích xuất đặc trưng hình ảnh
- Bộ giải mã văn bản (Text Decoder): Transformer decoder sinh ra chuỗi ký tự
- Tìm kiếm chùm (Beam search): Tìm kiếm chuỗi ký tự tối ưu với độ rộng chùm mặc định

Hậu Xử Lý Sau Khi Sinh Văn Bản của Transformer OCR

- Bộ mã hóa hình ảnh (Vision Encoder): Vision Transformer (ViT) trích xuất đặc trưng hình ảnh
- Bộ giải mã văn bản (Text Decoder): Transformer decoder sinh ra chuỗi ký tự
- Tìm kiếm chùm (Beam search): Tìm kiếm chuỗi ký tự tối ưu với độ rộng chùm mặc định

-Hậu xử lý và trực quan hóa

- Vẽ bounding boxes: Đánh dấu vùng phát hiện bằng đa giác màu xanh
- Gán nhãn văn bản: Hiển thị nội dung nhận dạng phía trên mỗi vùng
- Tạo heatmap: Hiển thị bản đồ nhiệt từ CRAFT
- Tổng hợp kết quả: JSON chứa thông tin đầy đủ

➤So sánh các OCR dựa vào chỉ số CER:

CER (Character Error Rate) không chỉ là một con số, mà là phép đo toàn diện về chất lượng nhận dạng văn bản ở cấp độ cơ bản nhất - cấp độ ký tự.

Đề tài: Xây dựng hệ thống bái đỡ xe thông minh ứng dụng thị giác máy tính

- Thử nghiệm trên 30 ảnh và ta được kết quả:

sst	Thực tế trả về	EASY trả về	Paddle trả về	Transformer trả về	CER eas	CER pad	CER tran
1 warehouse	(Eli	EASYHOUSE	AREHOUSE	WARE	1	0.1111	0.5556
2 VF6	VF6	VF6	VF6	VF6	0	0	0
3 VF6	Vfg	Vf6	Vf6	VE6	0.3333	0	0.3333
4 VINFAST	N	YIMEAST	VINEASTS	0.8571	0.4286	0.2857	
5 VF9	VF9	VF9	VF9	VF9	0	0	0
6 VF9	VFG	VF9	VF9	VF9	0.3333	0	0
7 i10	5710		OLIO	0.6667	1	1.3333	
8 Grand	OQANO	ONVEO	GRAND II	0.6	1	0.4	
9 L2	12	21	12	0.5	1	0.5	
10 GRAND	GRAND i10C	GRAND	GRAND NO	0.6	0	0.4	
11 i10	i10	i10	NO	0.3333	0	1	
12 VIOS	0	105	VIOS	1	1	0.25	
13 Expat	Expat	Expat	Expata	0	0	0.2	
14 Auto	uto	uto	uto	0.25	0.25	0.25	
15 CROSS	CROSS	ER	CROSS	0.2	0.8	0	
16 COROLLA	COROLLA	COROLLA	COROLIA	0	0	0.1429	
17 COROLLA	COROLLA	COROLLA	COROLIA	0	0.1429	0.1429	
18 LB	LEI	LB	LOL	1	0	1	
19 CROSS	CROSS	CROSS	CROSS	0	0	0	
20 EV710	FV710	EV710	EV710	0.2	0	0	
21 ford	Bordb	Ford	Bond	0.5	0	0.25	
22 RAN	NN	RAN	ORANG	1	0	0.6667	
23 G	G	5	G	0	1	0	
24 ER	R	R	IER	0.5	0.5	0.5	
25 Mitsubishi	Sishw	ceestettslin	mitsupshr	0.7	1.1	0.5	
26 Lancer	Lns	kc	lancer	0.8333	0.8333	0	
27 EL^NTR	L^NTR	N	ELANTRA	0.3333	0.8333	0.5	
28 ^			^	1	1	0	
29 ignis	ienis	ignis	ignis	0.2	0	0	
30 ignis	ignis	ignis	ignis	0	0	0	
				CER eas	CER pad	CER tran	
				CER trung bình của tất cả ảnh	0.4313	0.3666	0.307

Hình 2.18: Kết quả so sánh các OCR.

Kết luận: Nhờ vào các chỉ số CER vượt trội của OCR Transformer là 0.307 thấp hơn so với OCR Paddle là 0.3666 và OCR easy là 0.413

=>Chọn OCR Transformer làm mô hình nhận diện tên xe.

2.2.5. Trả về kết quả thông qua fuzzy matching:

2.2.5.1. Thu thập dữ liệu để trả về:

Dữ liệu được định dạng: Brand;Model;Year;Length (mm);Width (mm);Height (mm);Kerb Weight (kg);

Thu thập dữ liệu 10 phương tiện của 10 hãng tại Việt Nam.

Lưu ý:

+ Các phương tiện thông dụng để tăng hiệu quả nhận diện (vinfast, toyota,...)

Đề tài: Xây dựng hệ thống bái đỡ xe thông minh ứng dụng thị giác máy tính

+Nếu các xe có cùng model thì gom cân nặng lại. Ví dụ: VF 6 bản Base (Eco) nặng 1.550 kg và VF 6 bản Plus nặng 1.743 kg (trọng lượng không tải) thì cân nặng sẽ là 1550 -1743 kg vì khi không có sự khác biệt bên ngoài giữa hai xe mà do trọng lượng động cơ cũng như nội thất. VF 6 bản Base (Eco) nặng 1.550 kg và VF 6 bản Plus nặng 1.743 kg (trọng lượng không tải)

2.2.5.2. Hệ thống Fuzzymatcher trả về kết quả chính xác:

FuzzyMatcher là một hệ thống thông minh dùng để so khớp và tìm kiếm thông tin phương tiện từ văn bản OCR. Hệ thống sử dụng các thuật toán fuzzy matching để xử lý lỗi nhận dạng và tìm thông tin chính xác trong cơ sở dữ liệu.

➤Khởi tạo và Nạp Dữ Liệu:

Khởi Tạo Hệ Thống:

- Tải cơ sở dữ liệu từ file CSV chứa thông tin xe hơi
- Tự động trích xuất danh sách hãng xe và model
- Chuẩn bị dữ liệu cho quá trình tìm kiếm

Quy Trình Nạp Dữ Liệu:

1. Đọc file CSV với mã hóa UTF-8
2. Trích xuất thông tin: Hãng xe, Model, Năm sản xuất, Kích thước, Trọng lượng
3. Chuẩn hóa dữ liệu: Chuyển đổi sang chữ hoa, loại bỏ khoảng trắng thừa
4. Xây dựng cấu trúc dữ liệu tối ưu cho tìm kiếm

➤So khớp model với thuật toán fuzzy:

1. Chuẩn hóa văn bản đầu vào:

- Chuyển đổi sang chữ hoa
- Loại bỏ khoảng trắng và ký tự đặc biệt
- Chuẩn hóa định dạng

2. Tìm kiếm chính xác:

- So sánh trực tiếp văn bản đã chuẩn hóa với cơ sở dữ liệu
- Trả về kết quả nếu tìm thấy khớp hoàn toàn

3. So khớp tương đối (fuzzy matching):

- Sử dụng thuật toán SequenceMatcher để tính độ tương đồng
- So sánh với tất cả model trong cơ sở dữ liệu

Đề tài: Xây dựng hệ thống bái đồ xe thông minh ứng dụng thị giác máy tính

- Trả về model có độ tương đồng cao nhất

4. Xử lý từ khóa đặc biệt:

- Nhận diện các biến thể phổ biến của model
- Xử lý lỗi OCR phổ biến (VD: "O" thành "0", "I" thành "1")

Nguồn so khớp:

- Khớp hoàn hảo: 100% độ tương đồng
- Khớp tốt: Trên 80% độ tương đồng
- Khớp tương đối: 50-80% độ tương đồng
- Không khớp: Dưới 50% độ tương đồng

➤Tìm Kiếm Thông Tin Xe:

Tìm Kiếm Theo Hãng và Model:

1. Tìm kiếm chính xác:

- So khớp đồng thời hãng xe và model
- Trả về thông tin đầy đủ nếu tìm thấy

2. Tìm kiếm tương đối:

- Tìm theo hãng xe chính xác, model tương đối
- Tìm theo hãng xe tương đối, model chính xác

3. Tìm kiếm theo hãng (khi không có model):

- Trả về model đầu tiên của hãng xe đó
- Cung cấp thông tin cơ bản

Thông Tin Trả Về:

- Hãng xe (Brand)
- Model
- Năm sản xuất
- Kích thước (Dài, Rộng, Cao)
- Trọng lượng (Kerb Weight)

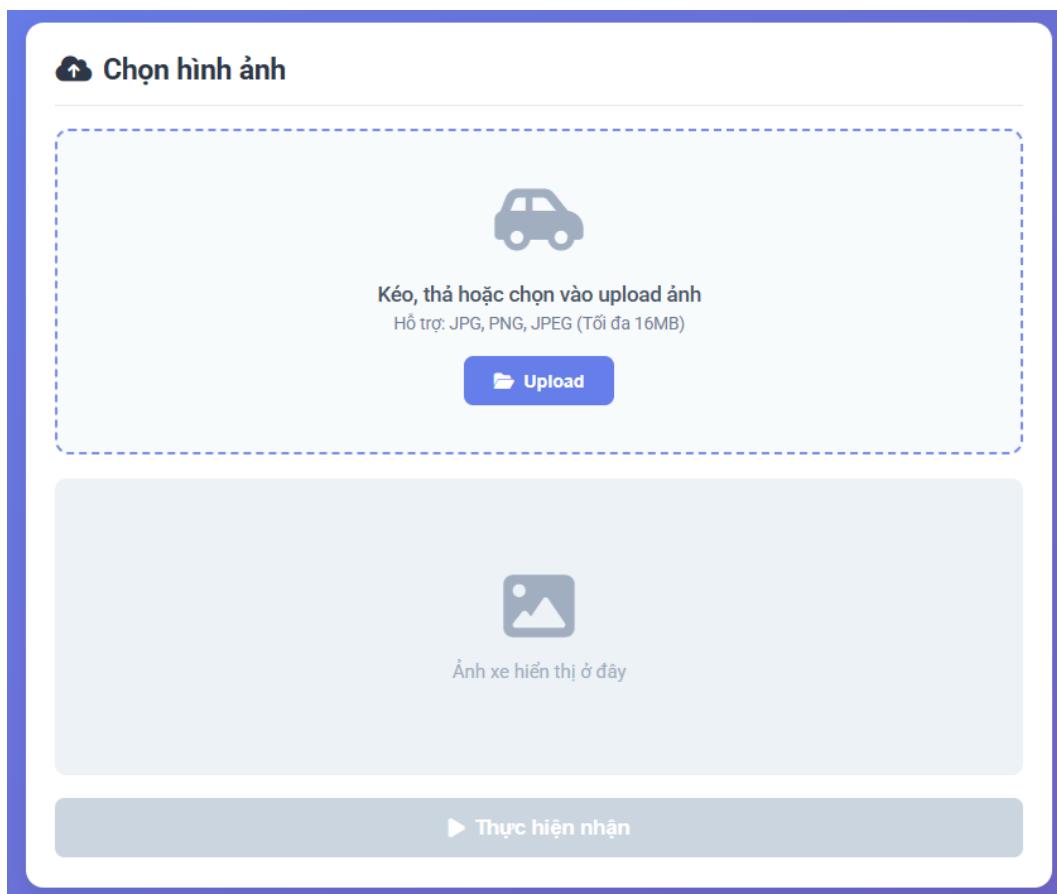
Kết luận: Dựa vào logo phương tiện lấy được từ YOLO và tên phương tiện lấy được từ OCR Transformer ta sử dụng Fuzzymatching để chuyển về tên chính xác thông qua file data.

CHƯƠNG 3. GIAO DIỆN , ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

3.1.Giao diện

3.1.1.Giao diện upload ảnh để nhận diện:

- Chọn ảnh tải lên để thực hiện nhận diện:



Hình 3.1: Giao diện nhận diện ảnh

3.3.2.Giao diện nhận kết quả:

- Nơi kết quả nhận diện trả về gồm các thông tin của phương tiện như tên hãng, tên xe, cân nặng, tầng đậu, vị trí đậu,....

Kết quả nhận

Thông tin phương tiện

Hàng (Trước khi chuẩn hóa): -

Hàng (Sau khi chuẩn hóa): -

Tên xe (Trước khi chuẩn hóa): -

Tên xe (Sau khi chuẩn hóa): -

Thông tin cân nặng và vị trí đậu

Cân nặng: - kg

Tầng: -

Vị trí: -

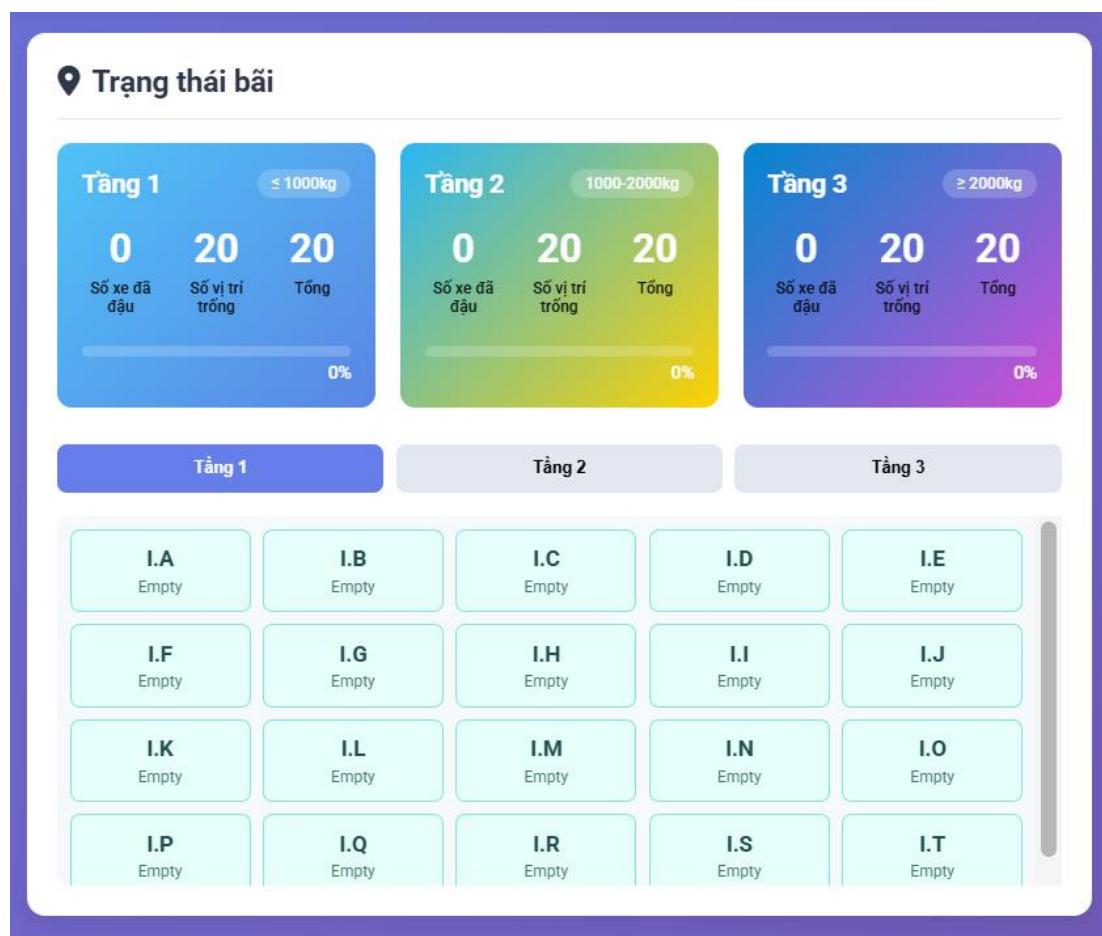
ID: -

✓ Xác nhận đã đỗ lại
➡ Phương tiện rời đi

Hình 3.2: Giao diện nhận kết quả

3.3.3. Giao diện trạng thái bãi đỗ:

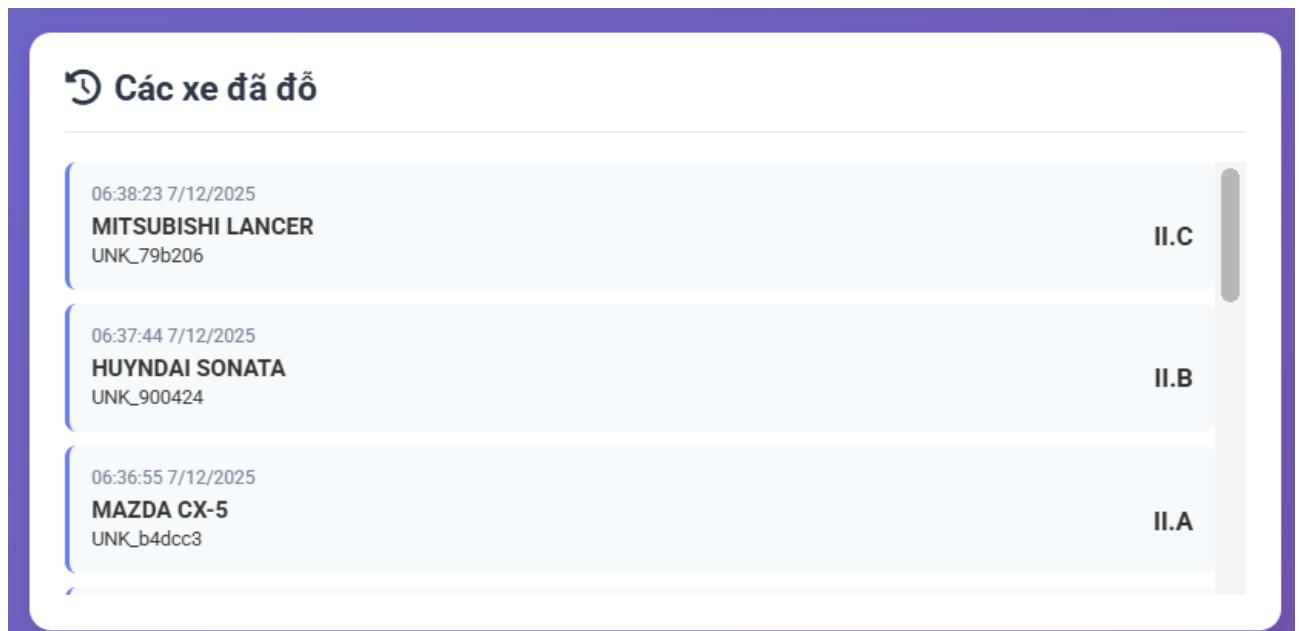
- Xem có bao nhiêu xe đang đậu trong bãi đỗ, tầng nào,



Hình 3.3: Giao diện nhận diện ảnh

3.3.4.Giao diện lịch sử đỗ xe của tất cả:

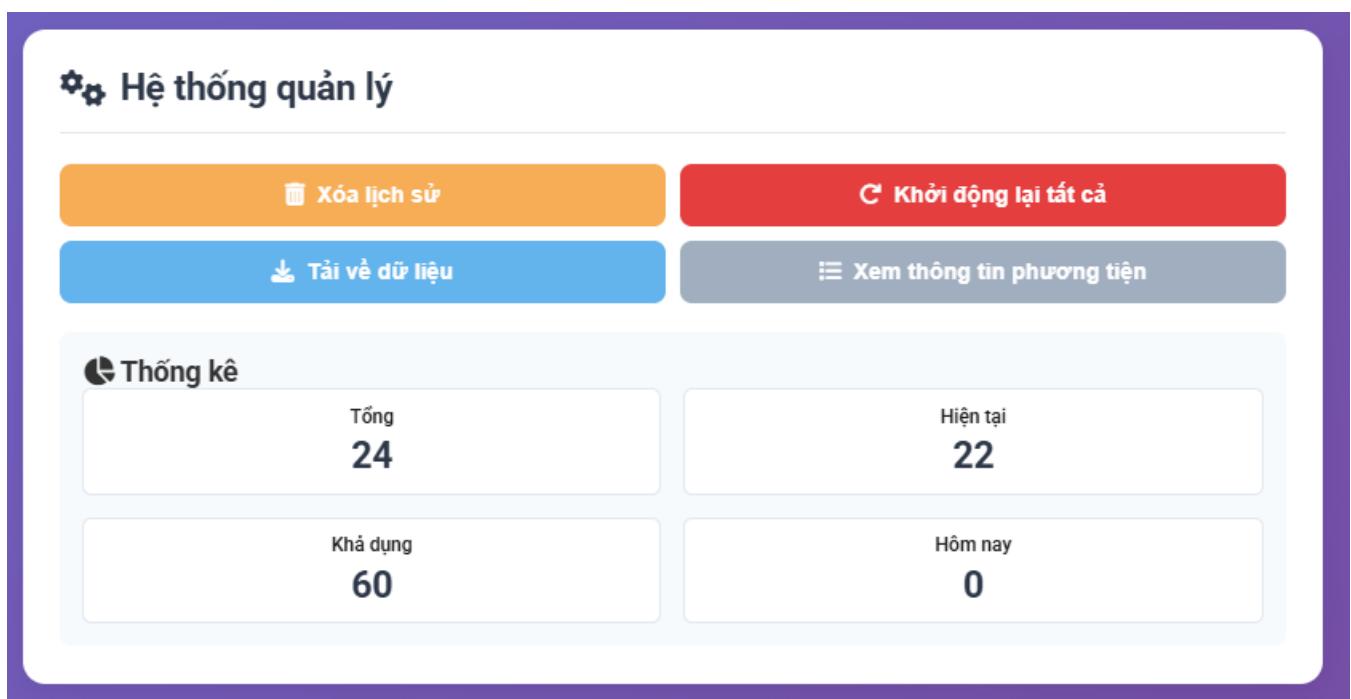
- Dùng để tra cứu thông tin đỗ xe,



Hình 3.4: Giao diện lịch sử đỗ xe

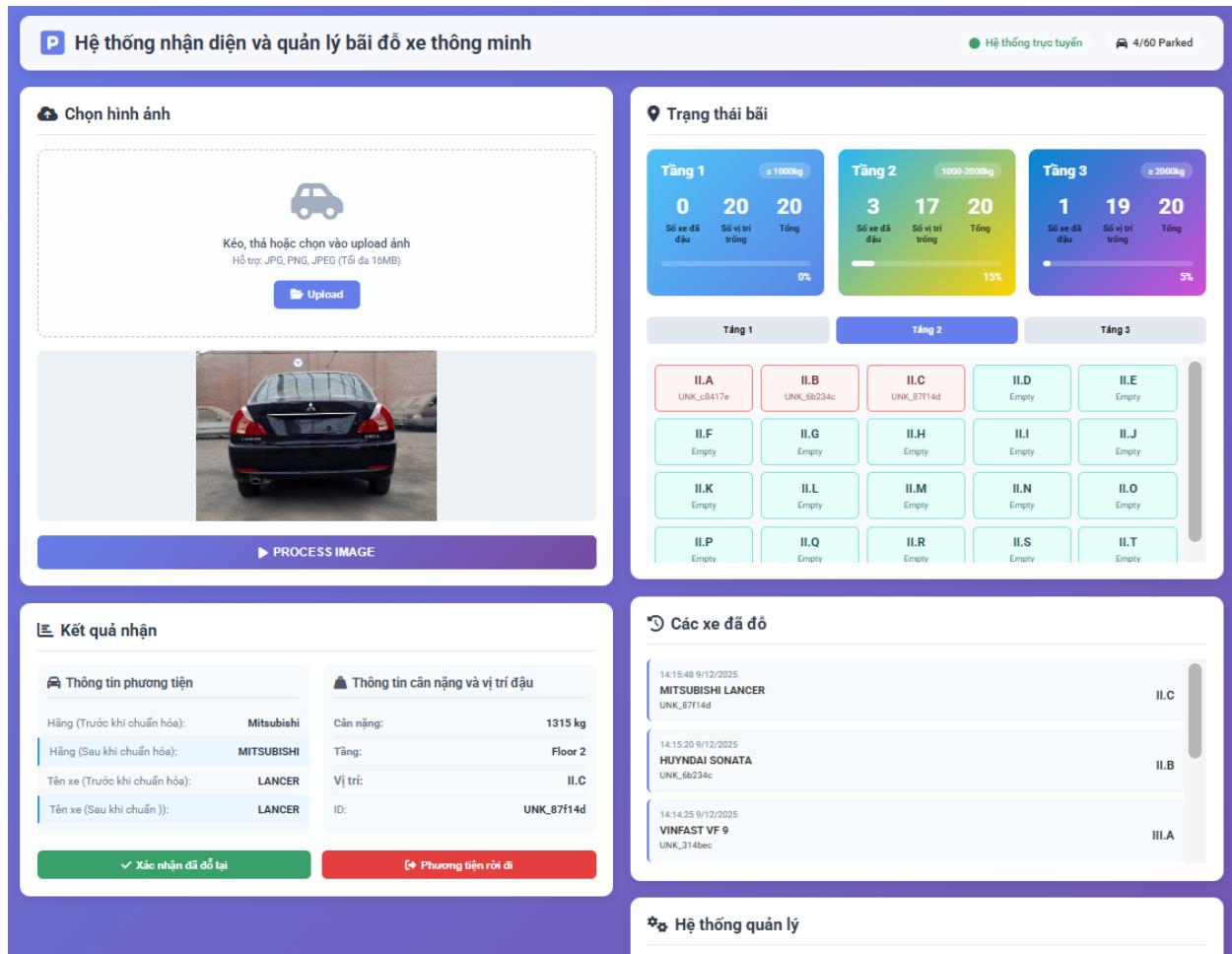
3.3.5.Giao diện quản lý:

- Dùng để thực hiện các tác vụ cơ bản như xóa lịch sử đỗ, thống kê



Hình 3.5: Giao diện quản lý

3.3.6.Giao diện tổng quát:



Hình 3.6: Giao diện tổng quát.

3.2.Đánh giá và hướng phát triển

3.2.1.Đánh giá R-CNN:

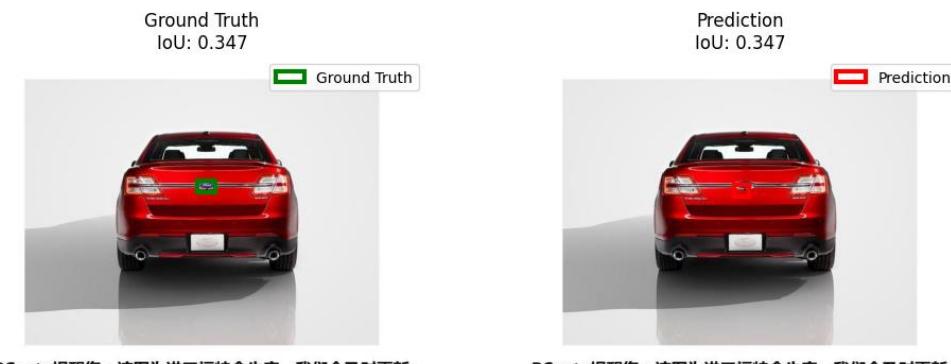
- Chỉ số kiểm thử khá tệ dù đã thay đổi và chỉnh sửa chỉ số nhiều lần

Testing on 106 images...

- KẾT QUẢ TEST TRÊN 106 ẢNH:
- Average IoU: 0.201
- Accuracy (IoU > 0.5): 13.21%
- Max IoU: 0.681
- Min IoU: 0.000

Đề tài: Xây dựng hệ thống bái đỡ xe thông minh ứng dụng thị giác máy tính

Test 1: 15421db9bf98dd.jpg.rf.bc024b07917760c8586e61cc2f309f34.jpg



Hình 3.7: Đánh giá và kiểm thử R-CNN.

Kết quả học được: Thiếu kiến thức về R-CNN chưa đủ kinh nghiệm để tự thiết kế một mô hình R-CNN hoàn chỉnh có thể nhận diện logo trong ảnh.
Tương lai có thể học hỏi và hoàn thiện mô hình có thể sử dụng.

3.2.2. Đánh giá CNN:

- Chỉ số kiểm thử khá ổn định qua nhiều lần chỉnh sửa.

KẾT QUẢ DỰ ĐOÁN 20 ẢNH ĐẦU TIÊN:

```
=====
Ảnh 1: 15421db9bf98dd.jpg.rf.bc024b07917760c8586e61cc2f309f34_000.jpg
    Thực tế: ford | Dự đoán: ford | Độ tin cậy: 0.999 | ✓ ĐÚNG
Ảnh 2: 2154245c593872.jpg.rf.e520ee909ae33dc46f2653f0fea6eb3_000.jpg
    Thực tế: ford | Dự đoán: ford | Độ tin cậy: 0.898 | ✓ ĐÚNG
Ảnh 3: 23a83c73722354.jpg.rf.cc71bb5ca79814e310d7ab9d093cb2f3_000.jpg
    Thực tế: ford | Dự đoán: ford | Độ tin cậy: 0.642 | ✓ ĐÚNG
Ảnh 4: b472a59a7029c3.jpg.rf.b961a162cd54c8f3b66f9cc609d9ee1f_000.jpg
    Thực tế: ford | Dự đoán: ford | Độ tin cậy: 0.543 | ✓ ĐÚNG
Ảnh 5: d0771c34b8dae0.jpg.rf.2fefbc6b6a2d78b9d2af381c603156f_000.jpg
    Thực tế: ford | Dự đoán: ford | Độ tin cậy: 0.912 | ✓ ĐÚNG
Ảnh 6: 177ace021e218d.jpg.rf.00d0548f9d04b475ab79dc1f2bf78649_000.jpg
    Thực tế: Honda | Dự đoán: Honda | Độ tin cậy: 0.992 | ✓ ĐÚNG
Ảnh 7: 7f94ddc79c9b46.jpg.rf.07471256d34005eb223128a0f14e14c2_000.jpg
    Thực tế: Honda | Dự đoán: Honda | Độ tin cậy: 0.850 | ✓ ĐÚNG
Ảnh 8: 803bda9ac93584.jpg.rf.f170e79cec3adfb91594d9f38fb09b77_000.jpg
    Thực tế: Honda | Dự đoán: Honda | Độ tin cậy: 0.812 | ✓ ĐÚNG
Ảnh 9: b0c6a3766c8373.jpg.rf.930b83bd97cb1b53be30783d0413f885_000.jpg
    Thực tế: Honda | Dự đoán: Honda | Độ tin cậy: 0.547 | ✓ ĐÚNG
Ảnh 10: dc29ea720092fe.jpg.rf.3155091415bd0e5b4f0b283ae828b5a5_000.jpg
    Thực tế: Honda | Dự đoán: Honda | Độ tin cậy: 0.952 | ✓ ĐÚNG
Ảnh 11: 16487bacaf5759.jpg.rf.88f2bf63b0e2dad7bc30e87f40dbf82f_000.jpg
    Thực tế: hyundai | Dự đoán: hyundai | Độ tin cậy: 0.699 | ✓ ĐÚNG
Ảnh 12: 16aca9c2391de7.jpg.rf.36505be7dc0bda3dcc4961b5a508ee8d_000.jpg
    ...
Ảnh 19: daa0b1d90b9916.jpg.rf.25b76a92959b1179375ab856f9dbcdfe_000.jpg
    Thực tế: KIA | Dự đoán: KIA | Độ tin cậy: 0.999 | ✓ ĐÚNG
Ảnh 20: f9c5e0af59cb2c.jpg.rf.5de6dab36e3358d2e63e94164c5f2858_000.jpg
    Thực tế: KIA | Dự đoán: KIA | Độ tin cậy: 0.933 | ✓ ĐÚNG
```

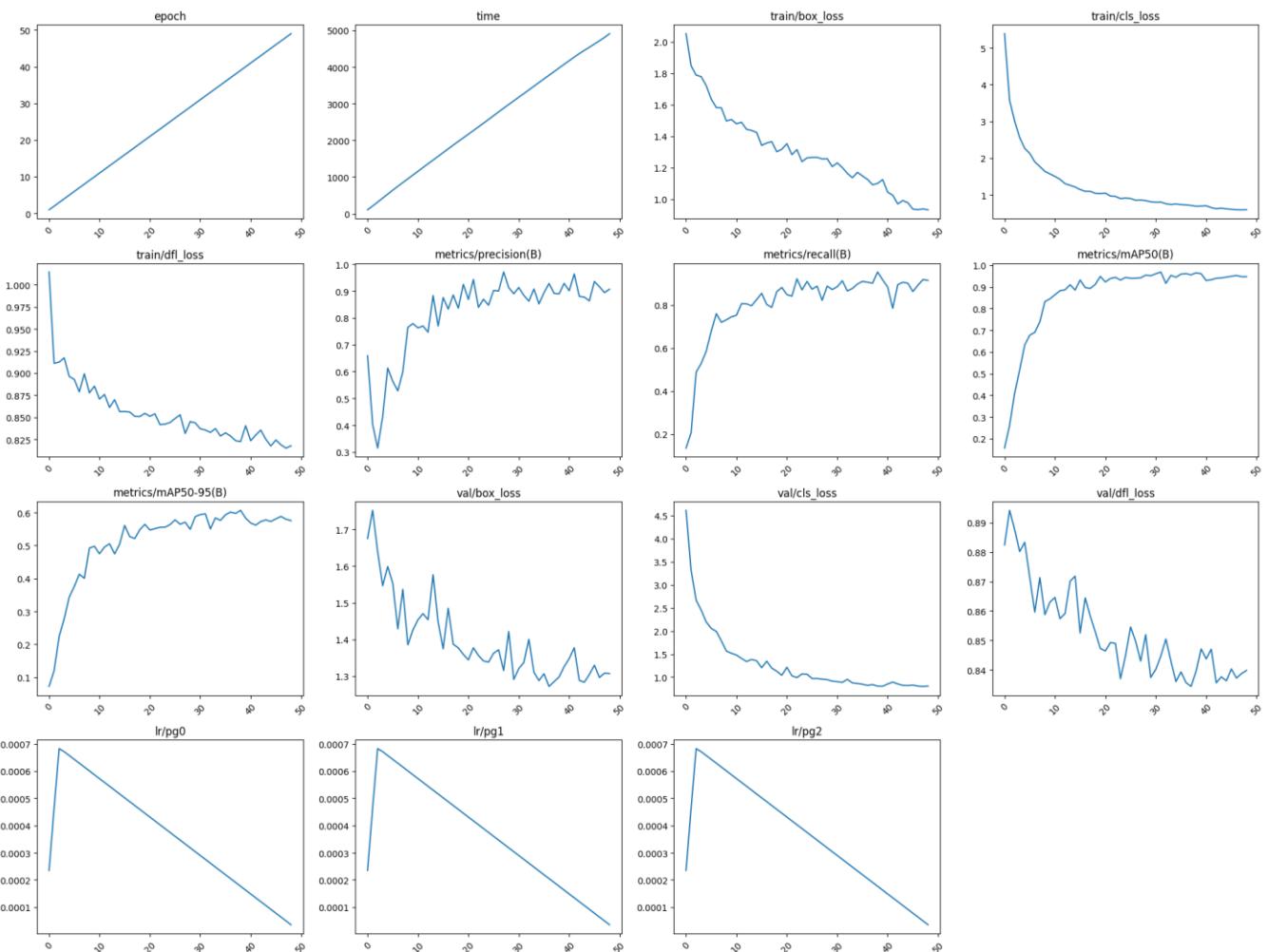
```
Found 51 images belonging to 10 classes.
2/2 [=====] - 1s 1!
Test loss: 0.3989
Test accuracy: 0.9020
```

Hình 3.8: Đánh giá và kiểm thử CNN.

Kết quả học được: Đã áp dụng được kiến thức đã học tại trường lớp để có thể chỉnh sửa và hoàn thiện mô hình CNN. Tương lai có thể nâng cấp, thiết kế mô hình để tăng độ chính xác, tốc độ ,....

3.2.3. Đánh giá YOLO:

- Các chỉ số đều ở mức ổn định và có thể sử dụng

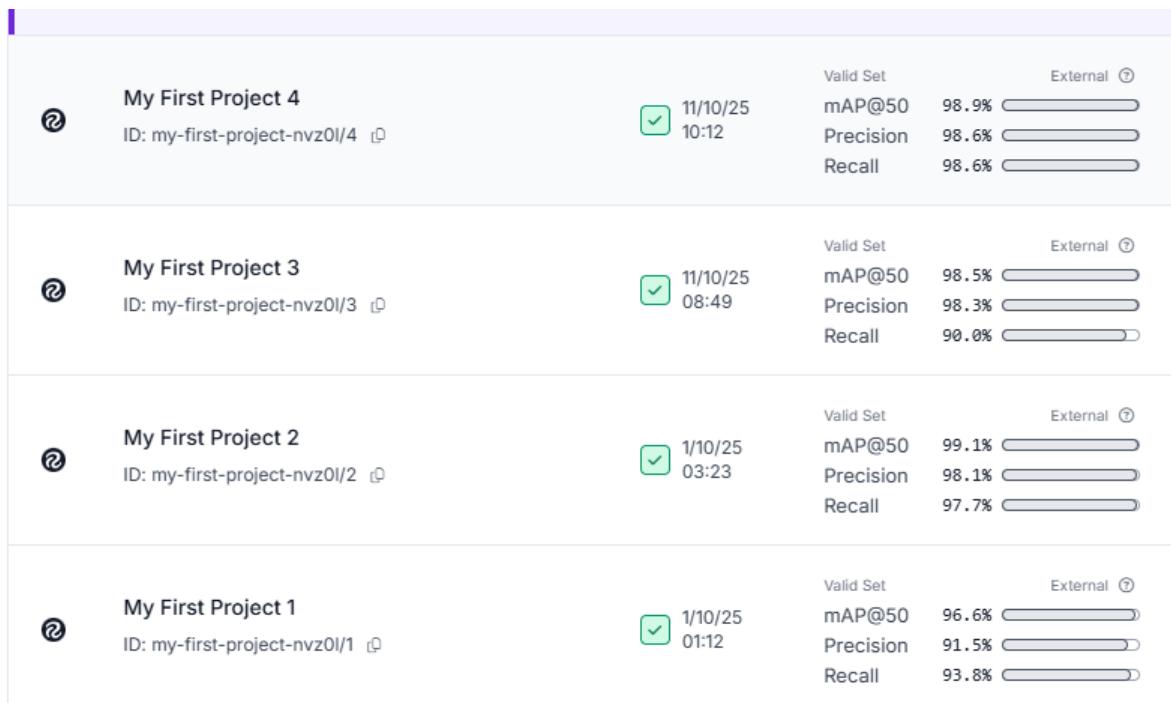




Hình 3.9: Dánh giá và kiểm thử YOLO.

Kết quả học được: Đã thành công huấn luyện mô hình YOLO có thể sử dụng cho đề tài quản lý bãi đỗ xe thông minh, tương lai nếu muốn cải thiện có thể nhờ các nền tảng hỗ trợ huấn luyện để tăng độ chính xác cho đề tài.

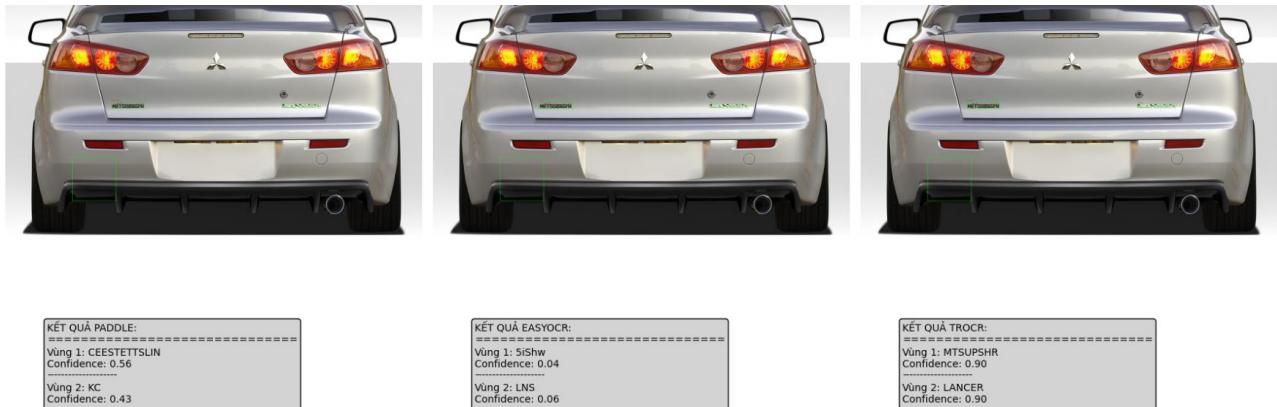
Ví dụ: Roboflow hỗ trợ huấn luyện mô hình YOLO:



Hình 3.10: Cải thiện YOLO bằng Roboflow.

3.2.3. Đánh giá OCR:

- Mỗi OCR đều có ưu và nhược điểm riêng tuy nhiên OCR Transformer lại vượt trội về chỉ số đánh giá CER nghĩa là Transformer có tỉ lệ nhận diện sai thấp nhất chỉ 0.307011 vậy nên OCR Transformer phù hợp với đề tài nhất.



Hình 3.11: Kiểm thử OCR.

Kết quả học được: Đã thành công so sánh các mô hình OCR để có thể tìm ra mô hình tốt nhất cho sử dụng để tài quản lý bái đỡ xe thông minh, tương lai nếu muốn cải thiện có thể thử thêm nhiều OCR thương mại để tăng độ chính xác như: Google Cloud Vision API, Amazon Textract, Microsoft Azure Computer Vision,....

3.2.4. Đánh giá Fuzzy matching:

- Fuzzy matching đã hoàn thành công ở mức ổn định.

1. Chuẩn hóa brand: 'yimfast'
→ Kết quả: 'vinfast'
2. Chuẩn hóa model: 've9'
→ Kết quả: 'vf 9'
3. Tìm xe với: brand='vinfast', model='vf 9'
Tìm thấy 1 xe phù hợp

```
-----
KẾT QUẢ:
-----
Brand:VinFast;Model:VF 9;Year:2024;Length (mm):5118;Width (mm):2254;Height (mm):1696;Kerb Weight (kg):2600-2866
  Brand: VinFast
  Model: VF 9
  Year: 2024
  Length (mm): 5118
  Width (mm): 2254
  Height (mm): 1696
  Kerb Weight (kg): 2600-2866
```

Hình 3.12: Kiểm thử Fuzzy matching.

Kết quả học được: Đã biết thêm phương pháp để hoàn thiện OCR, tương lai có thể thu thập thêm dataset để nhận diện được nhiều phương tiện hơn

III. KẾT LUẬN

Xây dựng thành công một trang web quản lý bãi đỗ xe thông minh với các chức năng chính bao gồm:

- **Nhận diện phương tiện bằng hình ảnh:** Đã thành công nhận diện được phương tiện và trả về được thông tin phương tiện để có thể đưa bãi đỗ
- **Quản lý bãi đỗ xe:** Hoàn thiện được các chức năng cơ bản, kiểm tra, thêm, xóa.

Tuy nhiên, đề tài còn có những hạn chế sau:

- **Hạn chế về dữ liệu:** Dữ liệu hình ảnh sử dụng để huấn luyện mô hình đa phần đến từ internet còn hạn chế về số lượng và độ đa dạng, đặc biệt là các ảnh thực tế, ảnh mờ.
- **K-CNN chưa hoàn thiện:** Mô hình K-CNN vẫn chưa thể hoạt động một cách tối ưu và ổn định
- **Hệ thống còn nhiều thiếu sót:** Trang web sơ sài, giao diện đơn giản, thiếu nhiều tính năng để trở thành bãi đỗ xe tự động như đăng ký tài khoản người dùng, quản lý khoản, chatbot hướng dẫn đỗ xe, thông tin thu phí, ...

TÀI LIỆU THAM KHẢO

- [1] Đỗ Thanh Nghị (Chủ biên), 2015, *Giáo trình Lập trình Web*, Nhà xuất bản Đại học Cần Thơ.
- [2] Đỗ Thanh Nghị (Chủ biên) & Phạm Nguyên Khang, 2022, *Giáo trình Khai thác dữ liệu với Python*, Nhà xuất bản Đại học Cần Thơ.
- [3] Đỗ Thanh Nghị & Phạm Nguyên Khang, 2012, *Giáo trình Nguyên lý máy học*, Nhà xuất bản Đại học Cần Thơ.
- [4] Steger and Carsten, Machine vision algorithms and application
- [5] Sonka and Milan, Image Processing, analysis and machine vision, Cengage Learning, 2014
- [6] Gary Rost Bradski and Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV library, O'Reilly, 2008.
- [7] Cluster analysis and data mining : an introduction, King, Ronald S., Dulles, Virginia: Mercury Learning and Information, 2015
- [8] Tom Mitchell, Machine Learning, McGraw Hill, 1997.
- [9] Kubat Miroslav, An Introduction to Machine Learning, Springer, 2017 E
- [10] Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Published 2014 by Cambridge University Press.