



TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

PBL4 – DỰ ÁN HỆ THỐNG THÔNG MINH

BÃI ĐỒ XE Ô TÔ HỖ TRỢ THU PHÍ TỰ ĐỘNG

Giảng viên hướng dẫn: TS. Ninh Khánh Duy

STT NHÓM: 3	
HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN ĐỒ ÁN
NGUYỄN VIỆT HOÀI BẢO	21N16C
NGUYỄN PHẠM NHẬT VỸ	21N16C
NGUYỄN ĐỨC DŨNG	21N16C
NGUYỄN HỮU HIỆP	21N16C

ĐÀ NẴNG, 01/2024

TÓM TẮT ĐỒ ÁN

Công nghệ thông tin ngày nay không chỉ là một xu hướng mà đã trở thành trụ cột không thể thiếu trong mọi lĩnh vực, từ doanh nghiệp đến giáo dục. Với sự hỗ trợ mạnh mẽ từ công nghệ thông tin, xã hội đã phát triển với tốc độ chóng mặt, và hệ thống tự động hóa đóng vai trò vô cùng quan trọng trong cuộc sống hiện đại.

Nhóm chúng em thực hiện đề tài bãi đỗ xe ô tô hỗ trợ tính phí tự động nhằm tạo ra một hệ thống quản lý và tối ưu hóa việc đỗ xe. Hệ thống này sẽ giúp tự động hóa bãi đỗ xe.

Để giải quyết vấn đề này, nhóm chúng em đã tạo ra một hệ thống thực hiện các chức năng chính như: theo dõi và quản lý bãi đỗ xe, giám sát việc xe ra vào; ứng dụng web giúp người dùng đăng ký biển số xe và thông tin cần thiết; ứng dụng phần mềm giúp quản lý bãi đỗ trong thời gian thực.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Kết quả
Nguyễn Viết Hoài Bảo	<ul style="list-style-type: none">- Thu thập dữ liệu- Gán nhãn dữ liệu – Roboflow- Xây dựng website – Django- Xây dựng thuật toán xoay ảnh – Python	<ul style="list-style-type: none">✓ Đã hoàn thành✓ Đã hoàn thành✓ Đã hoàn thành✓ Đã hoàn thành
Nguyễn Phạm Nhật Vỹ	<ul style="list-style-type: none">- Thu thập dữ liệu- Gán nhãn dữ liệu – Roboflow- Xây dựng website – Django- Database – SQLite	<ul style="list-style-type: none">✓ Đã hoàn thành✓ Đã hoàn thành✓ Đã hoàn thành✓ Đã hoàn thành
Nguyễn Đức Dũng	<ul style="list-style-type: none">- Tìm hiểu mô hình nhận diện- Xây dựng thuật toán nhận diện biển số xe – Python- Xây dựng thuật toán nhận diện ký tự - Python	<ul style="list-style-type: none">✓ Đã hoàn thành✓ Đã hoàn thành✓ Đã hoàn thành
Nguyễn Hữu Hiệp	<ul style="list-style-type: none">- Chuẩn bị, thiết kế và thực hiện phần cứng- Tạo hình cho mô hình	<ul style="list-style-type: none">✓ Đã hoàn thành✓ Đã hoàn thành

MỤC LỤC

1. GIỚI THIỆU.....	8
1.1. Hiện trạng vấn đề.....	8
1.2. Các vấn đề cần giải quyết	8
1.3. Đề xuất giải pháp tổng quan	8
2. GIẢI PHÁP.....	9
2.1. Tổng quan hệ thống	9
2.2. Giải pháp về phần cứng và truyền thông	10
2.2.1. Giải pháp phần cứng	10
2.2.2. Giải pháp truyền thông.....	13
2.3. Giải pháp phần mềm.....	14
2.3.1. Tổng quan về hệ thống nhận diện	14
2.3.2. Phát hiện và nhận diện biển số xe	15
2.3.3. Xoay biển số xe.....	16
2.3.4. Sắp xếp ký tự trên biển số xe	16
2.3.5. Phát hiện và nhận diện ký tự	17
2.3.6. Database	18
2.3.7. Framework Django	18
2.3.8. Use-case	20
2.3.9. Module Server.....	20
2.3.10. Web Server.....	21
3. KẾT QUẢ.....	24
3.1. Trực quan hóa tập dữ liệu	24
3.1.1. Tập dữ liệu cho phát hiện và nhận diện biển số xe	24
3.1.2. Tập dữ liệu cho phát hiện và nhận diện ký tự	26
3.2. Cơ sở lý thuyết đánh giá mô hình.....	27

3.3.	Kết quả phát hiện và nhận diện biển số xe theo mô hình YOLOv8	31
3.3.1.	Kết quả mô hình.....	31
3.3.2.	Kết quả thực tế	33
3.4.	Kết quả xoay ảnh biển số xe và sắp xếp ký tự theo mô hình YOLOv8.....	34
3.5.	Kết quả phát hiện và nhận diện ký tự	35
3.5.1.	Kết quả mô hình.....	35
3.5.2.	Kết quả thực tế	36
3.6.	Công nghệ và framework đã sử dụng	37
3.7.	Kết quả API	37
3.8.	Ứng dụng phần mềm	37
3.9.	Ứng dụng web.....	37
4.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	40
4.1.	Kết luận.....	40
4.2.	Hướng phát triển.....	41
5.	DANH MỤC TÀI LIỆU THAM KHẢO	41

DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ tổng quan hệ thống	9
Hình 2: Nguyên lý hoạt động của phần cứng.....	10
Hình 3: Sơ đồ tổng quan về hệ thống nhận diện	14
Hình 4: Phát hiện và nhận diện biển số.....	15
Hình 5: Phát hiện và nhận diện ký tự.....	17
Hình 6: Database diagram.....	18
Hình 7: Kiến trúc Django.....	19
Hình 8: Biểu đồ Use-case.....	20
Hình 9: Biển số tự thu thập và tập dataset trên mạng	24
Hình 10: Biểu đồ biến số 1 dòng và 2 dòng.....	25
Hình 11: Biểu đồ phân chia dữ liệu train, test, valid.....	25
Hình 12: Số lượng ảnh biển số có ký tự.....	26
Hình 13: Biểu đồ số lượng ký tự của từng chữ số	26
Hình 14: Biểu đồ số lượng ký tự chữ cái	27
Hình 15: Mean Average Precision (mAP) ^[4]	27
Hình 16: Công thức IoU ^[5]	28
Hình 17: Confusion Matrix ^[6]	28
Hình 18: Precision ^[6]	29
Hình 19: Công thức tính Precision ^[6]	29
Hình 20: Recall ^[7]	29
Hình 21: Công thức tính Recall ^[7]	30
Hình 22: Precision Recall Curve ^[8]	30
Hình 23: Confusion Matrix của mô hình phát hiện và nhận diện biển số xe	31
Hình 24: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 20 vòng cuối. 32	
Hình 25: Confusion Matrix Normalized của mô hình phát hiện và nhận diện ký tự	35
Hình 26: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 50 vòng cuối 35	
Hình 27: Giao diện ứng dụng phần mềm	37
Hình 28: Giao diện thông tin người dùng	38
Hình 29: Giao diện lịch sử gửi xe người dùng.....	39
Hình 30: Giao diện quản lý tài khoản	39
Hình 31: Giao diện quản lý lịch sử gửi xe	40

DANH MỤC BẢNG

Bảng 1: Đề xuất giải pháp tổng quan.....	8
Bảng 2: Linh kiện sử dụng.....	11
Bảng 3: Thống kê chi phí đồ án	13
Bảng 4: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 20 vòng cuối ..	32
Bảng 5: Kết quả mô hình phát hiện và nhận diện biển số.....	33
Bảng 6: Kết quả xoay ảnh biển số xe và sắp xếp ký tự.....	34
Bảng 7: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 50 vòng cuối ..	36
Bảng 8: Kết quả phát hiện và nhận diện ký tự	36

1. GIỚI THIỆU

1.1. Hiện trạng vấn đề

Trên thế giới, các hệ thống quản lý bãi đỗ xe đã có sự phát triển đáng kể. Công nghệ cảm biến, camera và phần mềm thông minh được tích hợp để tự động quản lý chỗ trống và điều khiển việc vào ra xe một cách hiệu quả. Tuy nhiên, tại Việt Nam, các hệ thống tương tự vẫn đang trong giai đoạn phát triển và chưa thực sự phổ biến. Các hệ thống hiện tại thường dựa vào thẻ cảm ứng hoặc mã QR, nhưng vẫn tồn tại những thách thức trong việc quản lý thông tin và tối ưu hóa sử dụng chỗ trống.

1.2. Các vấn đề cần giải quyết

Để triển khai hệ thống quản lý biển số xe, cần giải quyết một số vấn đề quan trọng. Đầu tiên là xây dựng mô hình để thực hiện các chức năng cơ bản của bãi đỗ. Tiếp theo, cần phát triển thuật toán nhận diện biển số xe. Cuối cùng, xây dựng ứng dụng phần mềm thực hiện chức năng: quản lý biển số xe theo thời gian thực; quản lý thông tin người dùng, bao gồm các chức năng kiểm tra, đăng ký, chỉnh sửa và xóa dữ liệu người dùng một cách linh hoạt.

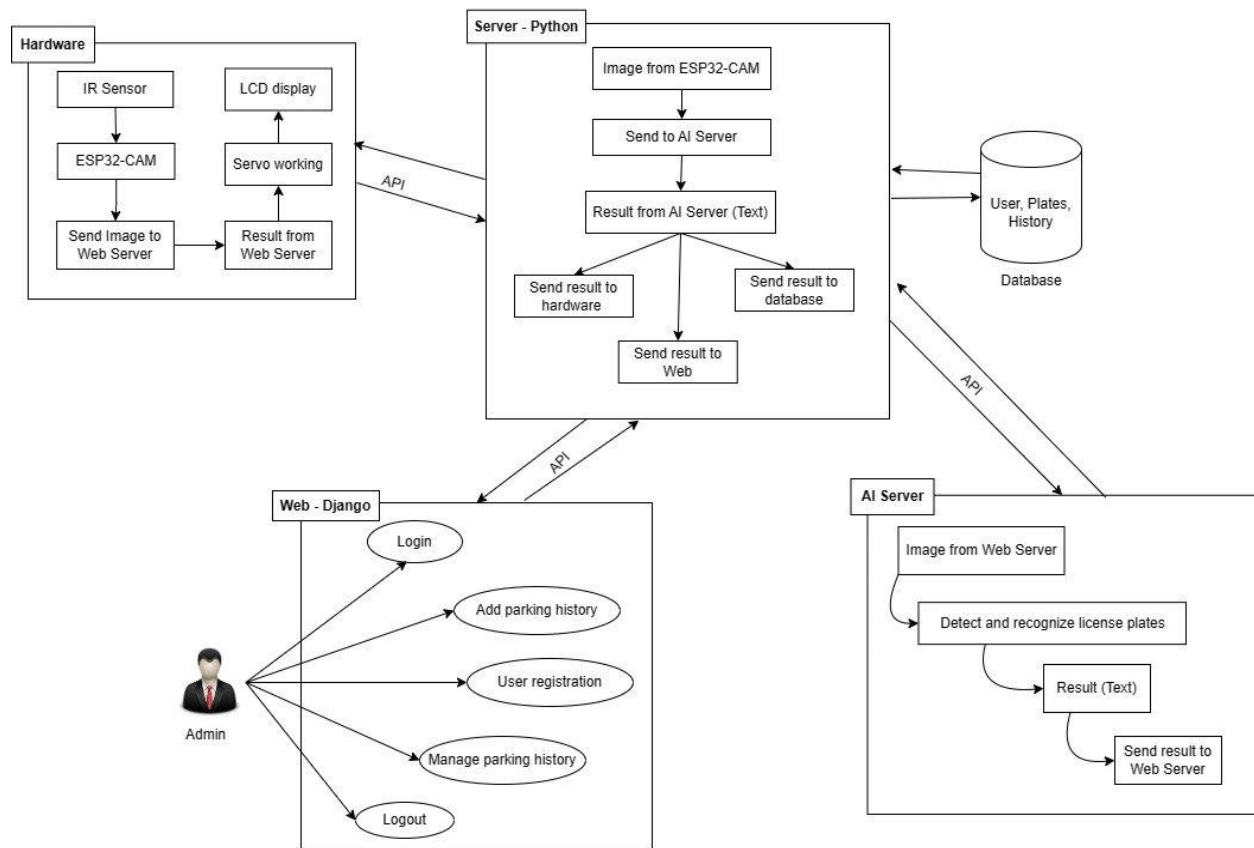
1.3. Đề xuất giải pháp tổng quan

Bảng 1: Đề xuất giải pháp tổng quan

Vấn đề	Giải pháp đề xuất
Phần cứng	Module wifi esp8266, Camera esp32, Servo.
Phát hiện biển số	Xây dựng và huấn luyện model phát hiện biển số. Huấn luyện trên google Colab (GPU)
Ứng dụng web	Ứng dụng Django để quản lý biển số xe theo thời gian thực và thông tin người dùng.

2. GIẢI PHÁP

2.1. Tổng quan hệ thống



Hình 1: Sơ đồ tổng quan hệ thống

Hệ thống bãi đỗ xe thông minh được chia thành 5 phần, mỗi phần đảm nhiệm một phần công việc khác nhau.

Vi điều khiển(Hardware): là nền tảng vật lý cho hệ thống, bao gồm các cảm biến, camera, động cơ,... có nhiệm vụ thu thập dữ liệu hình ảnh, gửi dữ liệu hình ảnh lên Server, nhận yêu cầu trả về và thực hiện điều khiển Barie.

Server: cung cấp các dịch vụ web cho hệ thống. Có nhiệm vụ giao tiếp giữa người quản lý và các phần khác trong hệ thống thông qua các APIs.

AI Server: thực hiện các thuật toán phát hiện và nhận diện biển số. Phần này có nhiệm vụ phân tích dữ liệu biển số xe ra - vào.

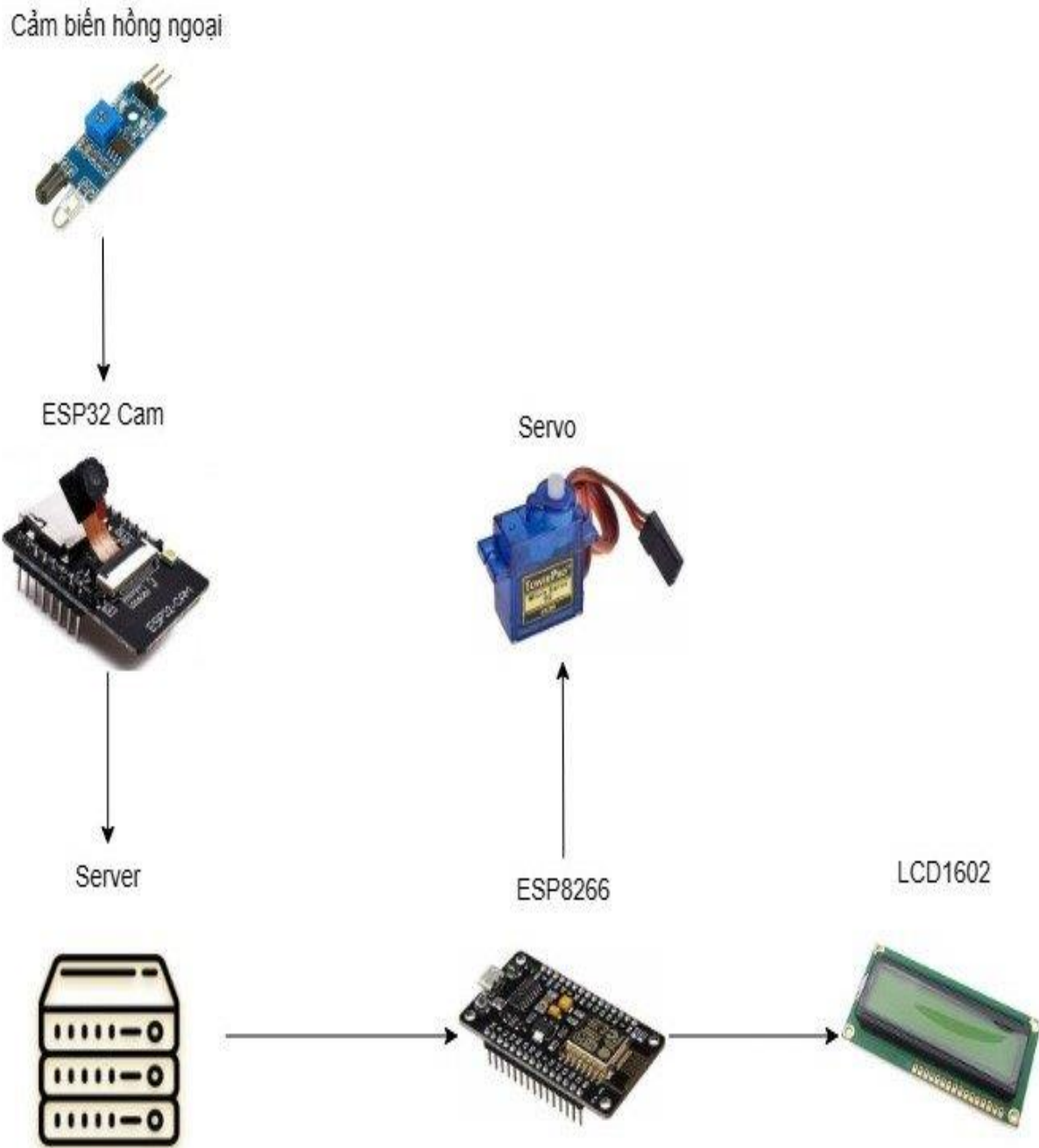
Database: lưu trữ dữ liệu về user, license plates, history. Phần này có nhiệm vụ lưu trữ và truy xuất dữ liệu một cách hiệu quả.

Web: có nhiệm vụ quản lý thông tin của người dùng, biển số xe, lịch sử ra vào.

2.2. Giải pháp về phần cứng và truyền thông

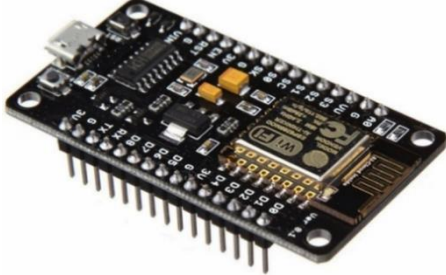

2.2.1. Giải pháp phần cứng


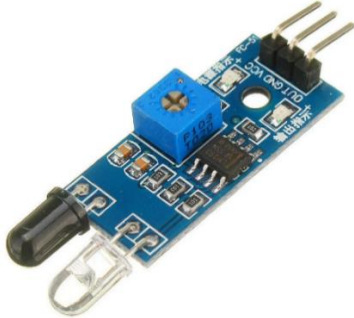

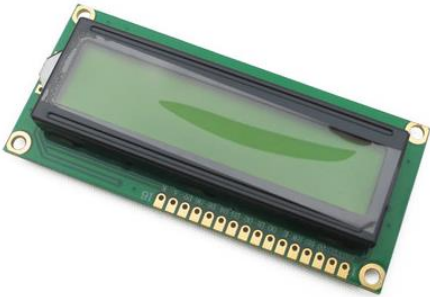
Nguyên lý hoạt động của phần cứng:



Hình 2: Nguyên lý hoạt động của phần cứng

Bảng 2: Linh kiện sử dụng

Tên linh kiện	Hình ảnh	Mô tả
Module wifi ESP8266		<p>Chip Wi-Fi: ESP8266EX.</p> <p>Điện áp làm việc: 3.3V DC.</p> <p>Được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.</p>
Camera ESP32		<p>Mạch tích hợp bộ xử lý chính là module ESP32 + Camera OV2640 .</p> <p>Điện áp hoạt động: 5V.</p> <p>Camera: Hỗ trợ camera OV2640 hoặc OV7670, xử lý định dạng JPEG, GRAYSCALE,...</p> <p>Được sử dụng trong các ứng dụng truyền hình ảnh, xử lý ảnh qua Wifi, Bluetooth hoặc các ứng dụng IoT.</p>

Đế nạp camera ESP32		Đế nạp chương trình và giao tiếp truyền dữ liệu với máy tính cho mạch ESP32-CAM.
Cảm biến hồng ngoại		<p>IC chính: LM393.</p> <p>Điện áp làm việc: 3.3 - 5VDC.</p> <p>Module cảm biến có thể được sử dụng để phát hiện hướng truyền có vật cản.</p> <p>Khoảng cách làm việc hiệu quả 2 ~ 5cm.</p>
Servo		Động cơ Servo SG90, góc quay 180 độ.
Màn hình LCD1602		<p>Điện áp hoạt động là 5 V.</p> <p>Kích thước: 80 x 36 x 12.5 mm.</p> <p>Chữ đen, nền xanh lá có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, dễ sử dụng thích hợp cho những người mới học và làm dự án.</p>

Bảng 3: Thống kê chi phí đồ án

Tên linh kiện	Số lượng	Đơn giá	Tổng
Module wifi ESP8266	2	55.000	110.000
Camera ESP32	2	145.000	290.000
Đế nạp ESP32	2	35.000	70.000
Cảm biến hồng ngoại	2	10.000	20.000
Màn hình LCD1602	2	75.000	150.000
Servo	2	25.000	50.000
Giấy, Dụng cụ làm mô hình		200.000	200.000

2.2.2. Giải pháp truyền thông

- Kết nối không dây:
 - + Camera kết nối với server thông qua module ESP32.
 - + Truyền tín hiệu cảm biến và nhận tín hiệu điều khiển động cơ thông qua module wifi ESP8266.
- REST API:

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác^[1].

REST (Representational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy^[1].

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. REST là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile,...) khác nhau giao tiếp với nhau^[1]:

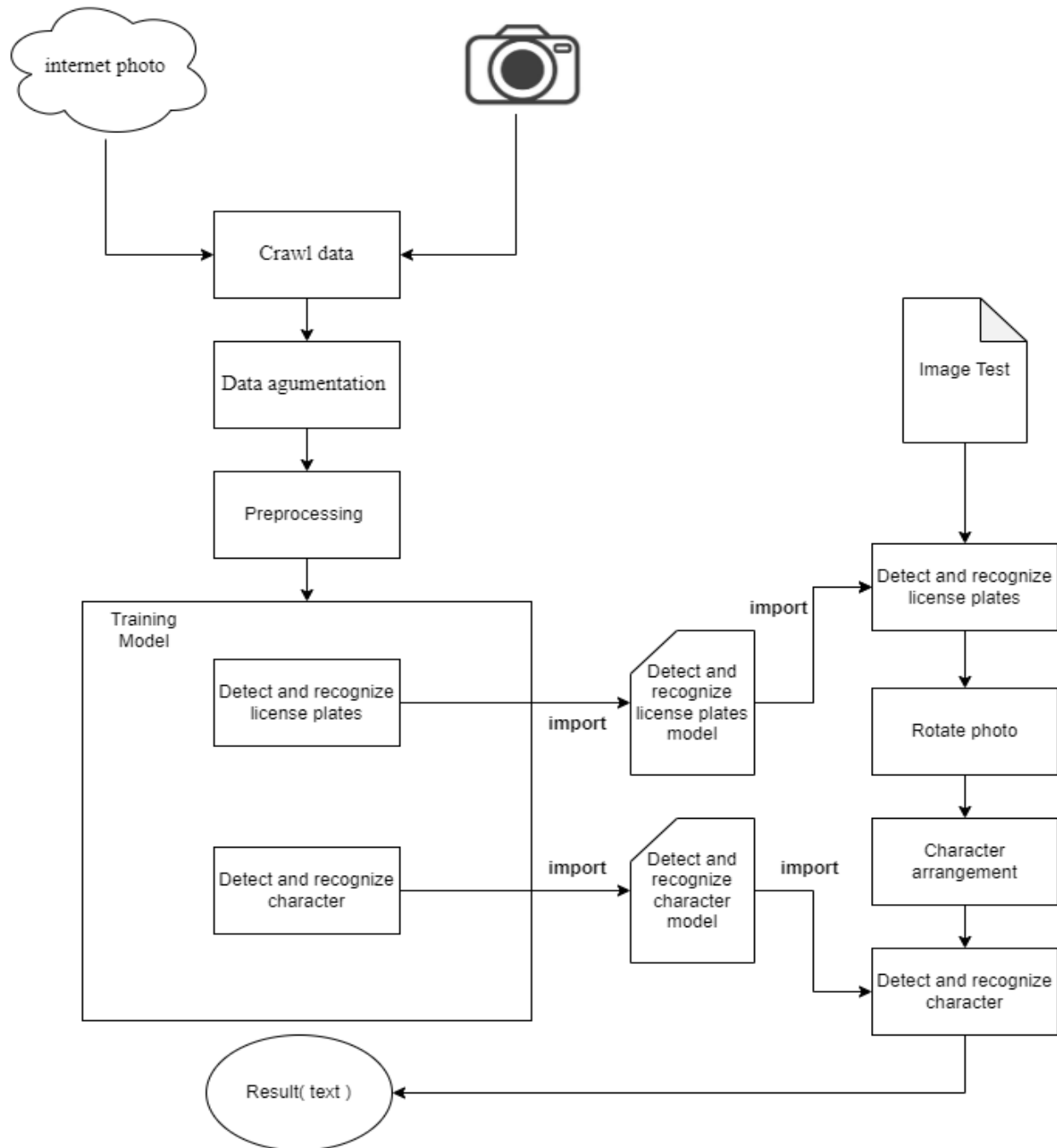
RESTful hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng ^[1]:

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xóa một Resource.

2.3. Giải pháp phần mềm

2.3.1. Tổng quan về hệ thống nhận diện



Hình 3: Sơ đồ tổng quan về hệ thống nhận diện

2.3.2. Phát hiện và nhận diện biển số xe

Phát hiện và nhận diện biển số xe (Detect and recognize license plates) là một module trong hệ thống dùng để phát hiện và nhận diện biển số có trong bức ảnh và đánh dấu bằng bounding box.

Sử dụng module chính là YOLOv8.



Hình 4: Phát hiện và nhận diện biển số

Mô hình YOLO (You Only Look Once) ^[2]: Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối. Ý tưởng của mô hình YOLO là chia hình ảnh thành một lưới ô vuông, mỗi ô vuông đại diện cho một khu vực tiềm năng có chứa đối tượng. YOLO sau đó dự đoán các thông tin sau cho mỗi ô vuông:

- Xác suất của việc có đối tượng trong ô vuông.
- Tọa độ của đối tượng trong ô vuông.
- Loại đối tượng trong ô vuông.

2.3.3. Xoay biến số xe

- **Vấn đề cần giải quyết**

Trong quá trình thực tế, phát sinh tình huống biến số xe bị nghiêng, gây khó khăn trong việc trích xuất chính xác thông tin từ biến số. Để giải quyết vấn đề này, nhóm em đã phát triển một thuật toán nhằm xác định góc nghiêng của biến số và thực hiện việc xoay ảnh để đưa về dạng chuẩn.

- **Thuật toán xoay biến số**

- Đầu tiên, xác định các đường thẳng gần trung tâm ảnh và chọn những đường thẳng có góc tạo thành với trục nhỏ hơn 40 độ.
- Tiếp theo, tính trung bình góc tạo bởi các đường thẳng đã chọn.
- Áp dụng ma trận xoay với góc đã tính được để xoay ảnh và thu được ảnh đã được điều chỉnh góc nghiêng.

- **Lý do sử dụng thuật toán**

Vì tập dữ liệu ảnh không quá phức tạp và yêu cầu việc xử lý nhanh chóng cùng với độ phức tạp thấp, thuật toán xoay ảnh được tích hợp và phần mềm quản lý xe trong thời gian thực, do đó nhóm em lựa chọn một giải pháp có tốc độ xử lý nhanh và độ phức tạp thấp để áp dụng.

2.3.4. Sắp xếp ký tự trên biến số xe

- **Vấn đề cần giải quyết**

Trong trường hợp biến số xe có một dòng, các ký tự đã được sắp xếp theo thứ tự chính xác.

Trong trường hợp biến số xe có hai dòng, áp dụng thuật toán để xác định vị trí của từng ký tự để lưu trữ thông tin một cách chính xác.

- **Thuật toán sắp xếp ký tự trên biến số xe**

- Xác định hộp chứa của từng ký tự trên biến số xe.

- Tìm phương trình đường thẳng đi qua tâm của hai hộp chứa khoảng cách xa nhất.
- Duyệt qua từng hộp chứa và kiểm tra xem tâm của hộp chứa đó có nằm trên đường thẳng với độ lệch là 20px.
- Phân biệt biển số là một dòng hay hai dòng.
- Tạo một ảnh hoàn chỉnh mới với các ký tự được sắp xếp theo thứ tự từ trái sang phải

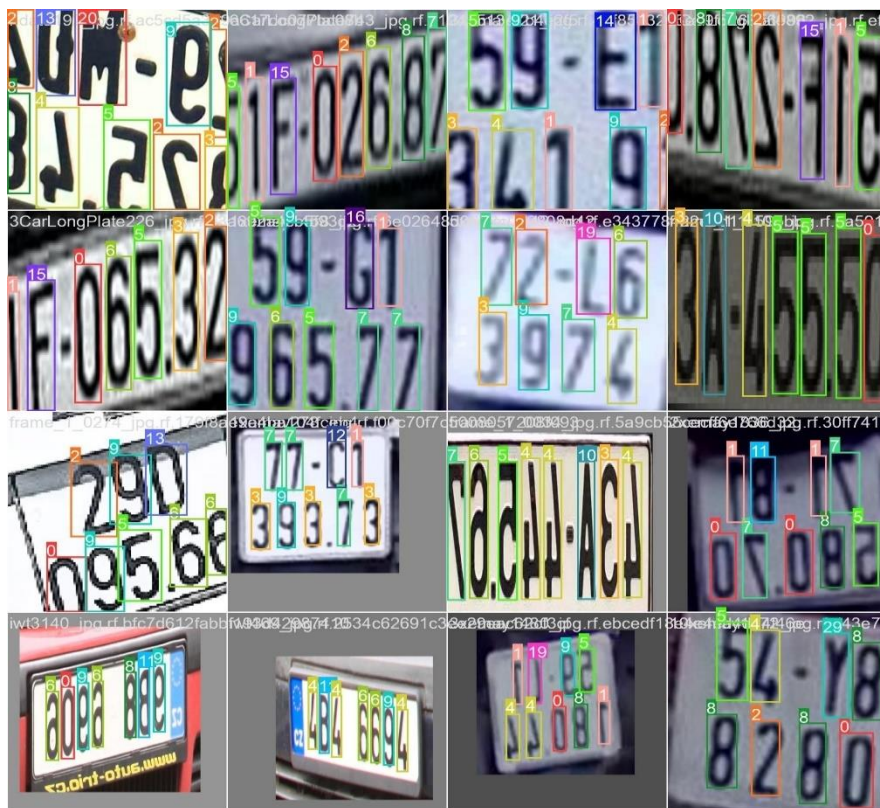
- **Lý do sử dụng thuật toán**

Qua việc áp dụng hai thuật toán này, chúng em đã có khả năng xử lý hiệu quả vấn đề xoay biển số và sắp xếp chữ cái trên biển số xe, từ đó cải thiện đáng kể quá trình trích xuất thông tin và quản lý dữ liệu biển số xe một cách chính xác và hiệu quả.

2.3.5. Phát hiện và nhận diện ký tự

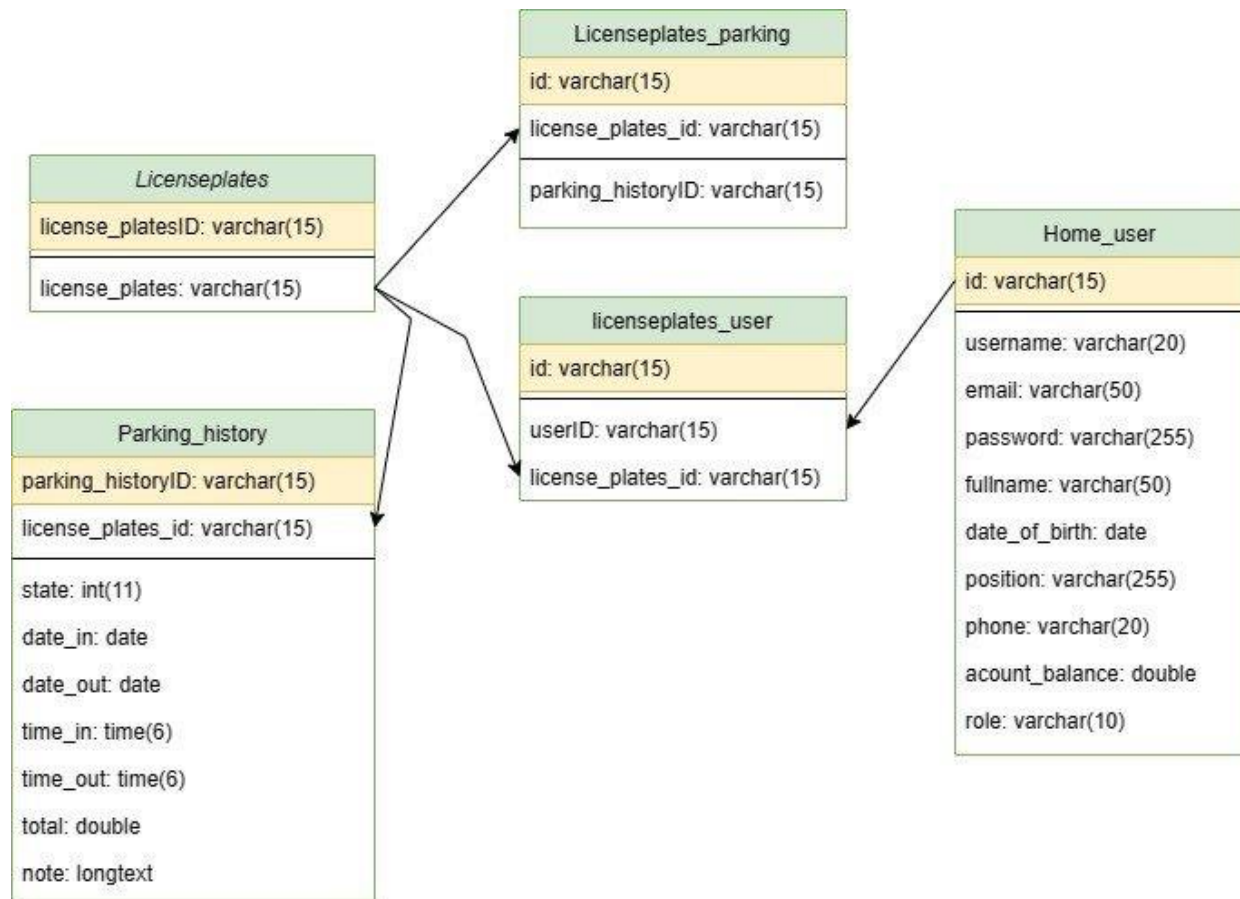
Phát hiện và nhận diện ký tự (Detect and recognize character) là một module trong hệ thống dùng để phát hiện và nhận diện ký tự có trong bức ảnh và đánh dấu bằng bounding box.

Sử dụng module chính là YOLOv8.



Hình 5: Phát hiện và nhận diện ký tự

2.3.6. Database



Hình 6: Database diagram

2.3.7. Framework Django

Xây dựng ứng dụng web bằng Django quản lý thông tin người đăng ký biển số xe và hỗ trợ thu phí cho bãi đỗ xe. Gồm các tác vụ cơ bản như nhập, sửa, xóa thông tin biển số và thông tin cá nhân người đăng ký.

Đối tượng sử dụng: admin có thẩm quyền, user.

Django là một framework bậc cao của Python có thể thúc đẩy việc phát triển phần mềm thần tốc và clean, thiết kế thực dụng. Được xây dựng bởi nhiều lập trình viên kinh nghiệm, Django tập trung lớn những vấn đề phát triển Web, bạn có thể phát triển trang web của bạn mà không cần xây dựng từ những căn bản. Đặc biệt nó free và open source^[3].

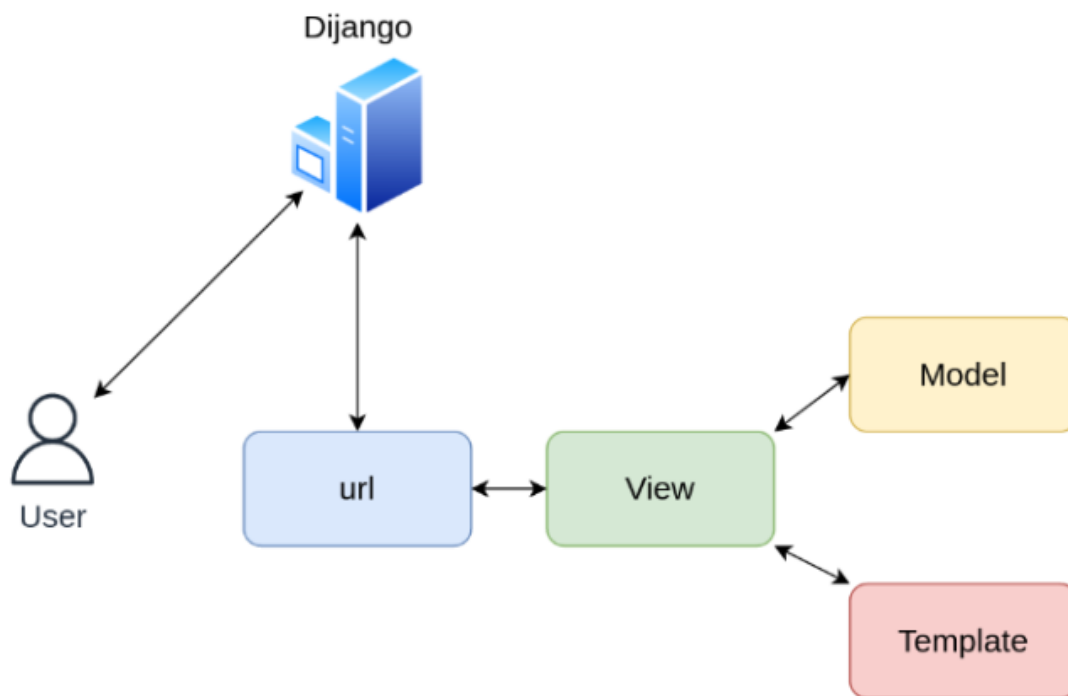
Nhưng lợi thế của Django:

- Đa năng: Django có thể được dùng để xây dựng hầu hết các loại website.

- Bảo mật: Django cung cấp các kỹ thuật bảo mật được xây dựng sẵn trong framework để giúp các nhà phát triển bảo vệ website của họ.
- Dễ maintain: code Django được viết theo nguyên tắc thiết kế và pattern có thể khuyến khích ý tưởng bảo trì và tái sử dụng code.
- Tính linh động: Django được viết bằng Python, nó có thể chạy đa nền tảng.

Kiến trúc:

Django mặc định đã áp dụng mô hình MVC ngay từ đầu, mô hình đó được diễn đạt như sau:

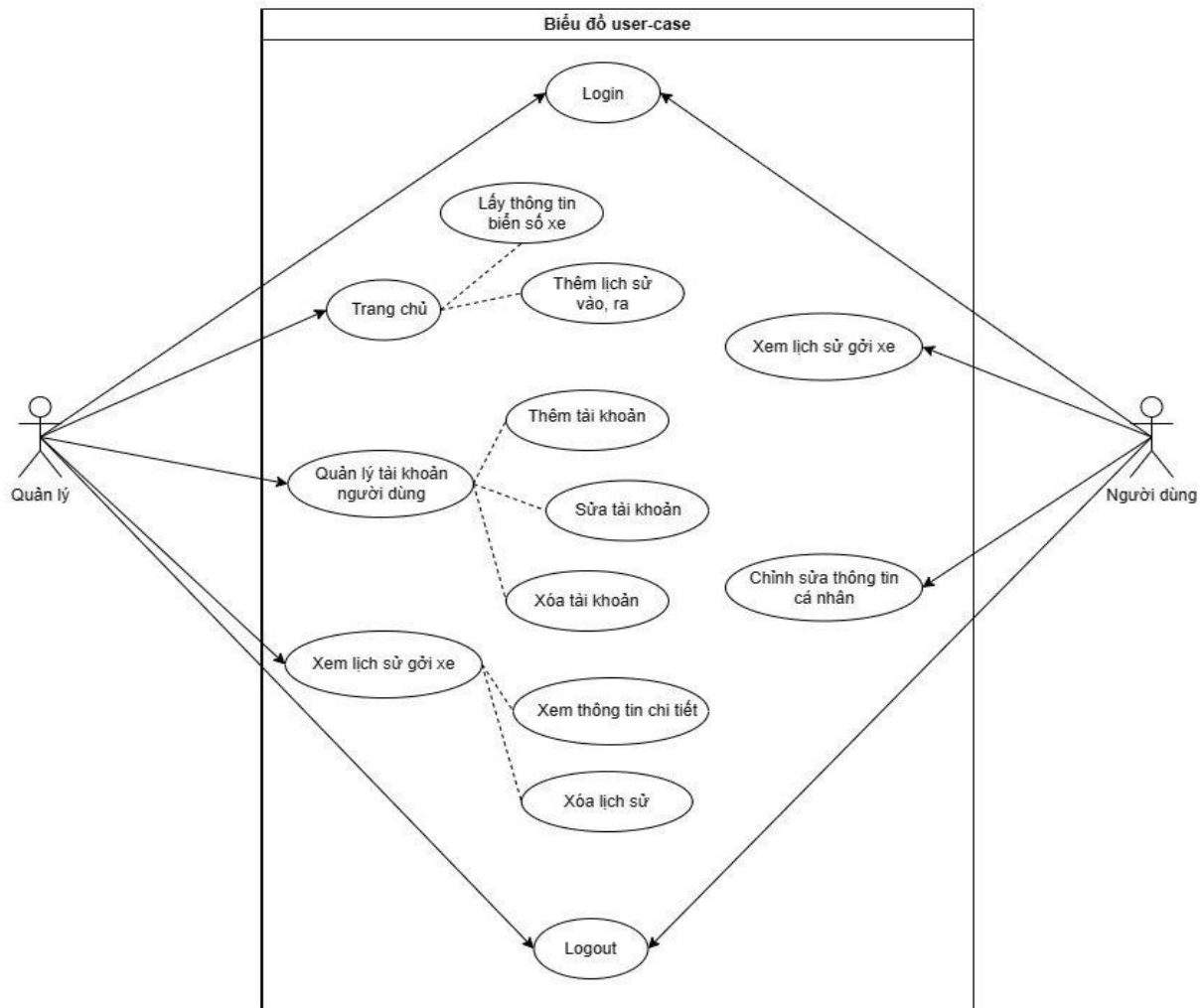


Hình 7 : Kiến trúc Django

- Models: Chính là Model trong MVC, đây chính là nơi ta thiết kế ra những table cho database, từ đó Django ORM đã cung cấp những phương thức xử lý, nghiệp vụ lên database.
- Template: Chính là View trong MVC, là những template ta thiết kế ra cho trang web.

- View: Đây là Controller trong MVC, trong view có các function xử lý khi có request từ người dùng.

2.3.8. Use-case



Hình 8: Biểu đồ Use-case

2.3.9. Module Server

Xây dựng module server nhằm cung cấp các API cho web server có thể kết nối đến và điều khiển phần cứng của hệ thống.

- <http://espIP/servoIn>
 - Method: GET
 - Parameters: None
 - Function: Điều khiển mở servo lối vào

- <http://espIP/servoOut>
 - Method: GET
 - Parameters: None
 - Function: Điều khiển mở servo lỗi ra
- <http://espIP/hongNgoaiIn>
 - Method: GET
 - Parameters: None
 - Function: Lấy thông tin cảm biến khi phát hiện xe vào
- <http://espIP/hongNgoaiOut>
 - Method: GET
 - Parameters: None
 - Function: Lấy thông tin cảm biến khi phát hiện xe ra
- <http://lcdIP/lcd/carIn?carIdIn=<str:carId>>
 - Method: GET
 - Parameters: <str:carID>
 - Function: Lấy id xe vào
- <http://lcdIP/lcd/carOut?carIdOut=<str:carId>&cost=<str:cost>>
 - Method: GET
 - Parameters: <str:carId>, <str:cost>
 - Function: Lấy id xe ra và phí gửi xe
- <http://lcdIP/Slot>
 - Method: GET
 - Parameters: None
 - Function: Lấy số lượng slot còn trống

2.3.10. Web Server

Xây dựng API cho server sẽ sử dụng django framework, restful API. Các API chia theo các chức năng của cùng một đối tượng.

Account:

- <http://localhost:8000/login>
 - Method: POST
 - Parameters: None
 - Function: Đăng nhập
- <http://localhost:8000/register>
 - Method: POST
 - Parameters: None
 - Function: Đăng kí tài khoản mới

User:

- <http://localhost:8000/indexUser>
 - Method: GET
 - Parameters: <str: userID> “id người dùng”
 - Function: Lấy thông tin người dùng
- <http://localhost:8000/updateUser>
 - Method: PUT
 - Parameters: <str: userID> “id người dùng”
 - Function: Cập nhật thông tin người dùng
- <http://localhost:8000/historyUser>
 - Method: GET
 - Parameters: <str: userID> “id người dùng”
 - Function: Lấy thông tin lịch sử gửi xe của người dùng

Admin:

- <http:// esp32 ip:port/stream>
 - Method: GET
 - Parameters: None
 - Function: Lấy nguồn stream từ camera

- [http:// esp32_ip:port/capture](http://esp32_ip:port/capture)
 - Method: GET
 - Parameters: None
 - Function: Lấy frame từ luồng video MJPEG
- <http://localhost:8000/managerStaff>
 - Method: GET
 - Parameters: None
 - Function: Lấy danh sách người dùng
- <http://localhost:8000/managerStaff/register>
 - Method: POST
 - Parameters: None
 - Function: Đăng kí tài khoản mới
- <http://localhost:8000/managerStaff/User>
 - Method: PUT
 - Parameters: <str: userID> “id người dùng”
 - Function: Cập nhật thông tin người dùng
- <http://localhost:8000/managerStaff/delete>
 - Method: POST
 - Parameters: <str: userID> “id người dùng”
 - Function: Xóa người dùng
- <http://localhost:8000/historyStaff>
 - Method: GET
 - Parameters: None
 - Function: Lấy lịch sử gửi xe
- <http://localhost:8000/historyStaff/history>
 - Method: GET
 - Parameters: <str: historyID> “id lịch sử”
 - Function: Lấy thông tin chi tiết lịch sử gửi xe

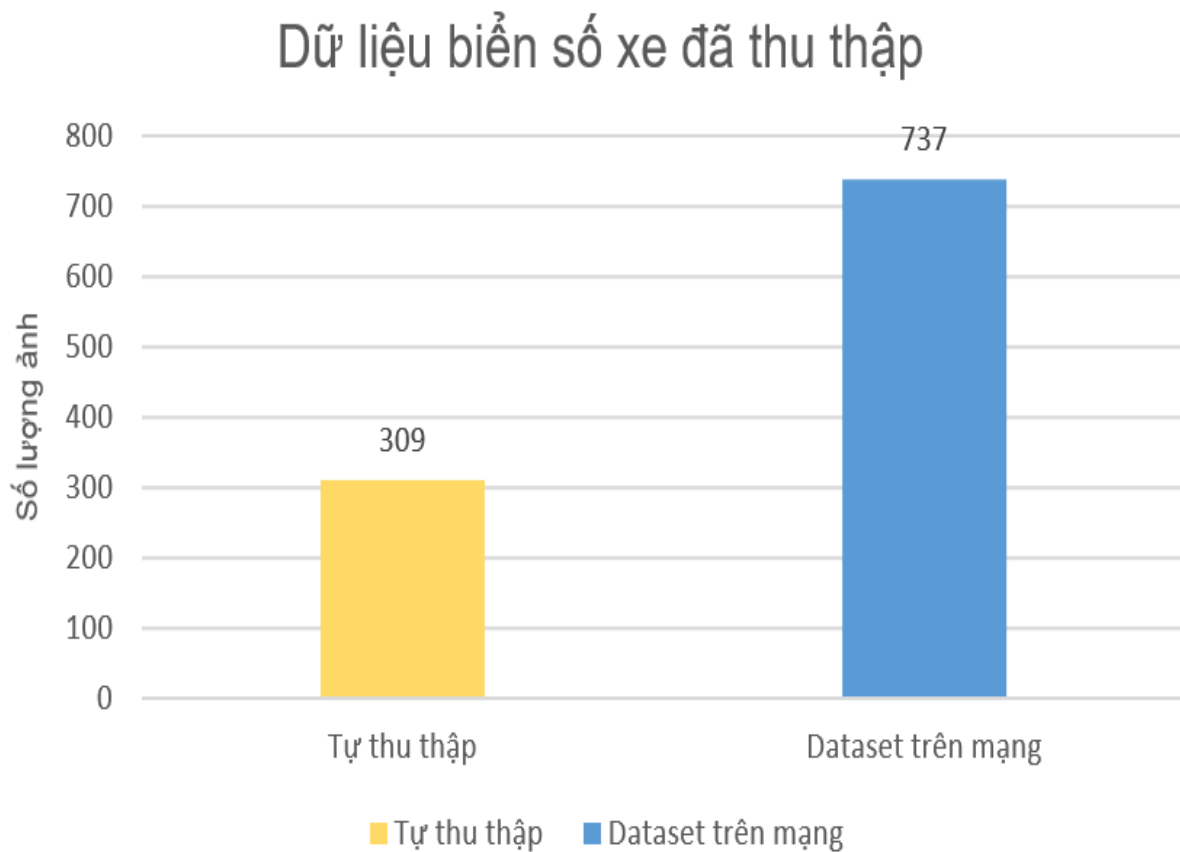
- <http://localhost:8000/historyStaff/delete>
 - Method: POST
 - Parameters: <str: historyID> “id lịch sử”
 - Function: Xóa lịch sử gửi xe

3. KẾT QUẢ

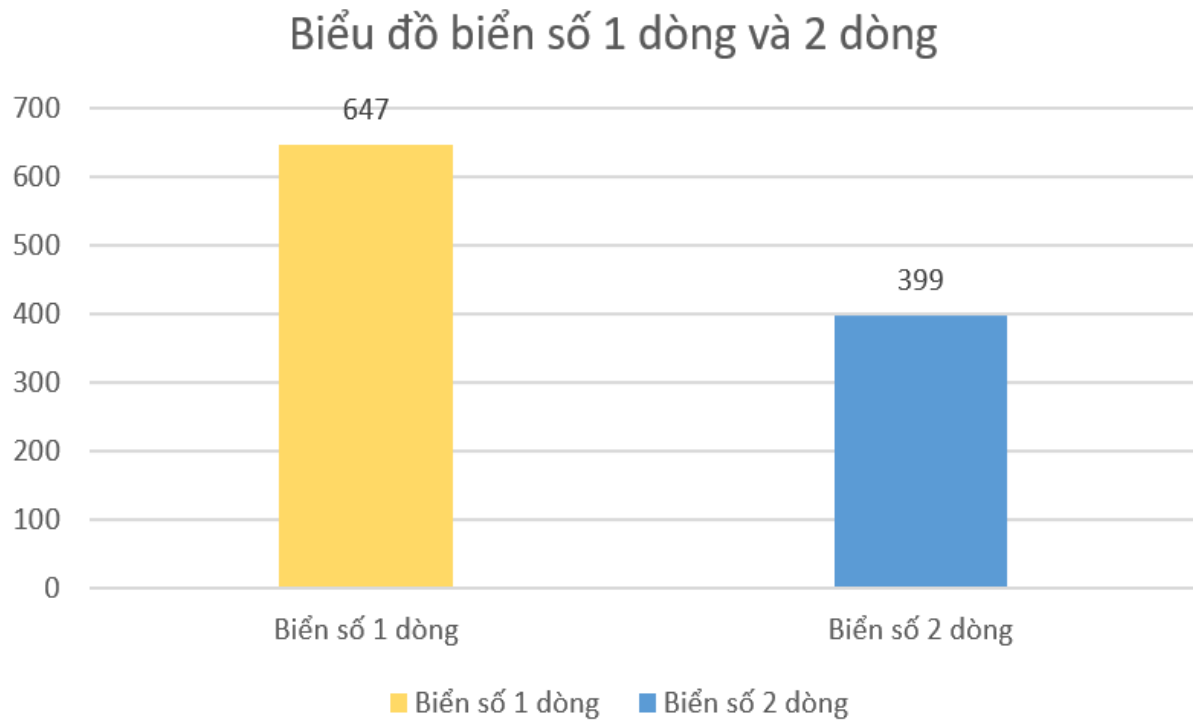
3.1. Trực quan hóa tập dữ liệu

3.1.1. Tập dữ liệu cho phát hiện và nhận diện biển số xe

- Số lượng dữ liệu: 1046 (ảnh)



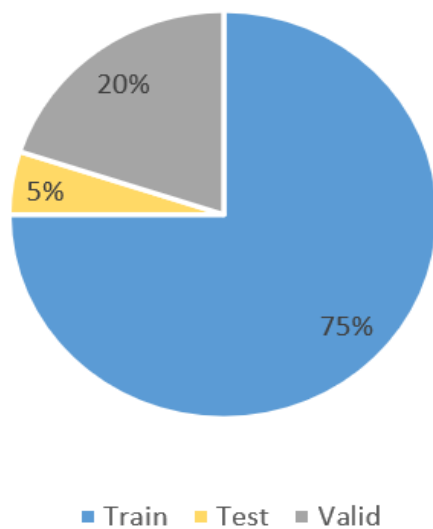
Hình 9: Biểu số tự thu thập và tập dataset trên mạng



Hình 10: Biểu đồ biến số 1 dòng và 2 dòng

- Phân chia tập dữ liệu:

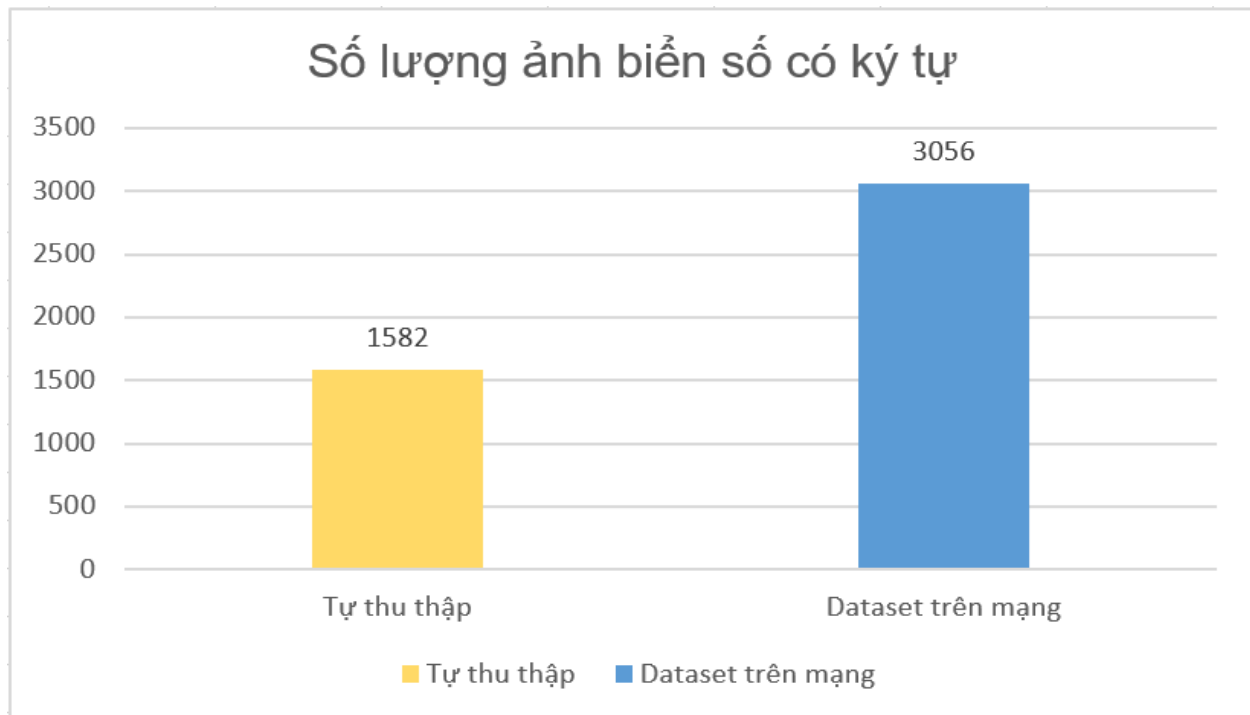
Tỷ lệ tập train, test, valid



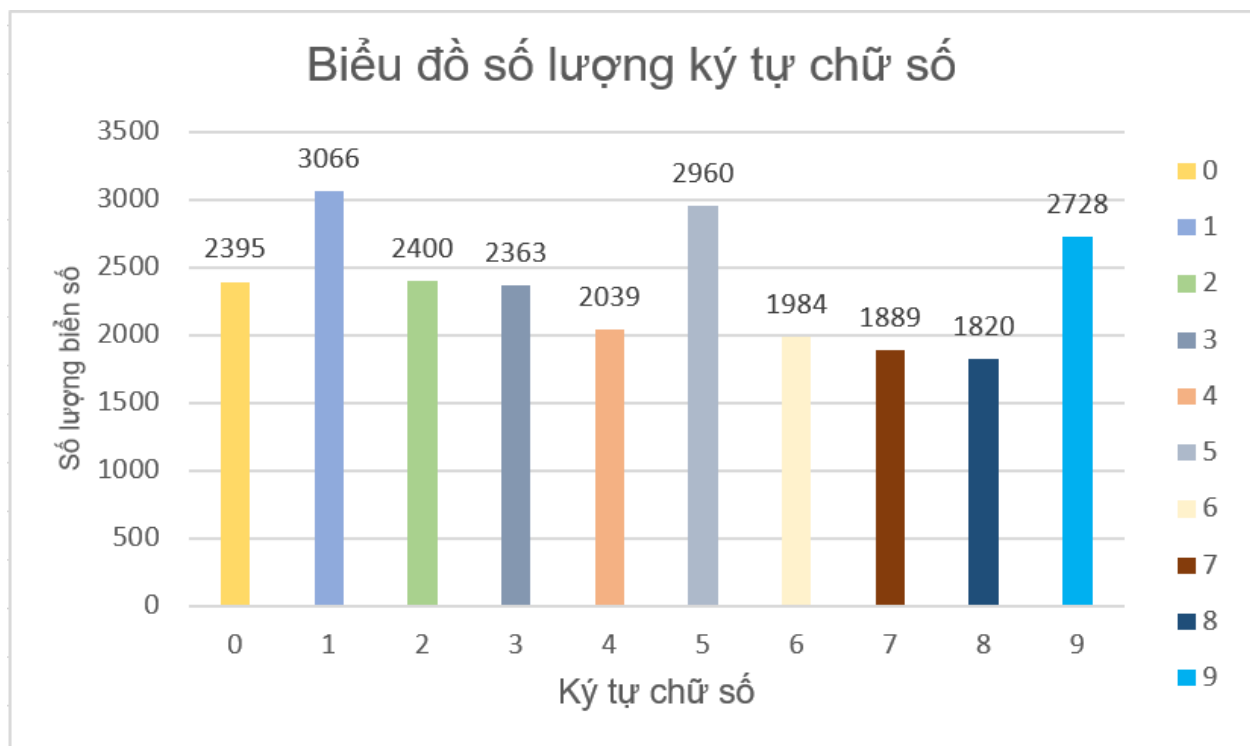
Hình 11: Biểu đồ phân chia dữ liệu train, test, valid

3.1.2. Tập dữ liệu cho phát hiện và nhận diện ký tự

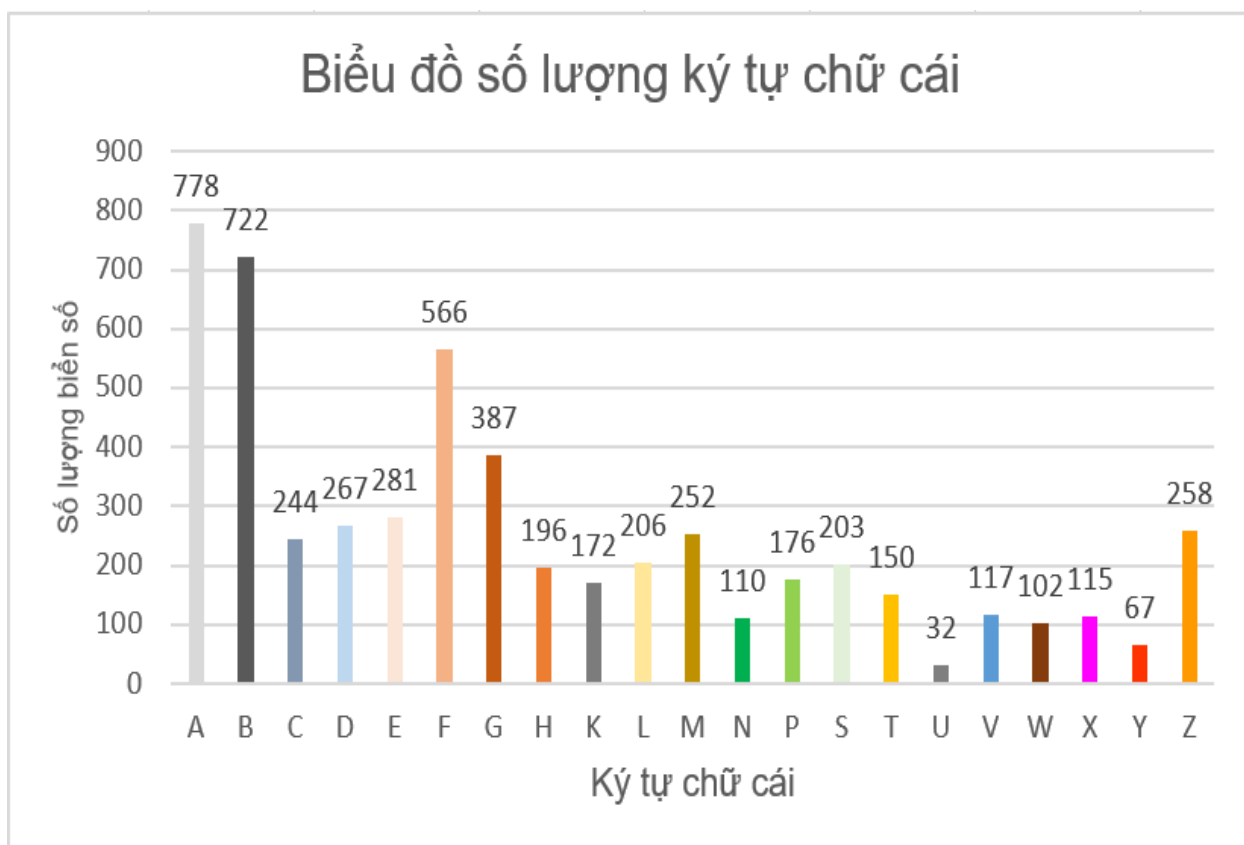
- Số lượng dữ liệu: 4638 (ảnh)



Hình 12: Số lượng ảnh biển số có ký tự



Hình 13: Biểu đồ số lượng ký tự của từng chữ số

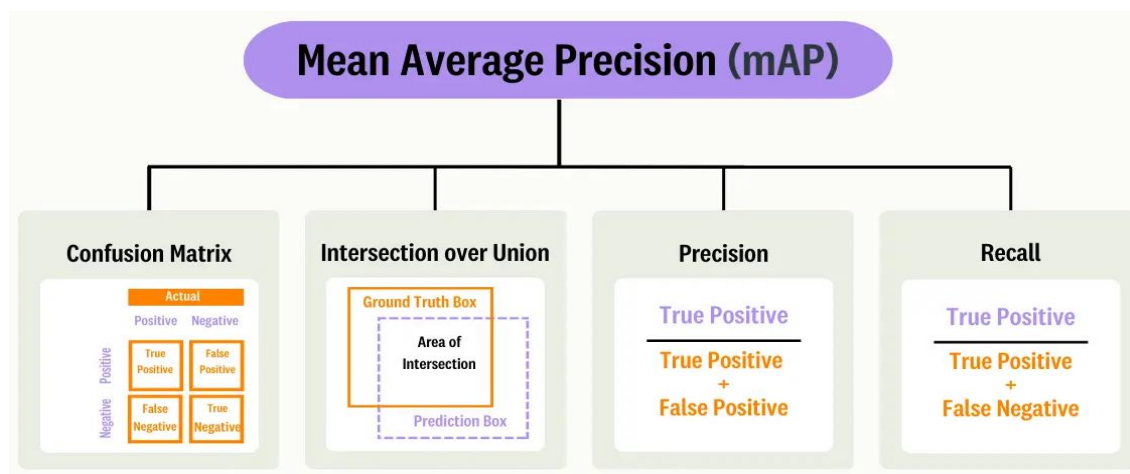


Hình 14: Biểu đồ số lượng ký tự chữ cái

3.2. Cơ sở lý thuyết đánh giá mô hình

Để tiến hành đánh giá hiệu suất mô hình, chúng em sử dụng mAP.

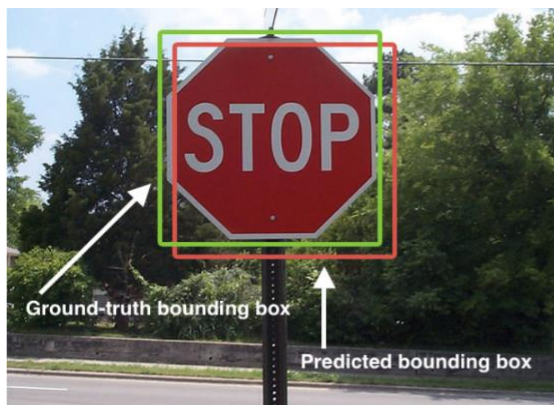
mAP sẽ là một tiêu chí đo lường của các model Object Detection. Để thu được mAP chúng ta cần đi qua tính toán lượt các phần sau:



Hình 15: Mean Average Precision (mAP) ^[4]

- IoU:

Iou là tỷ lệ giữa phần giao và phần hợp của vùng dự đoán và vùng đối tượng thật.



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Hình 16: Công thức IoU^[5]

Trong đó:

- Area of Overlap là diện tích vùng giao nhau giữa predicted bounding box và ground truth.
- Area of Union bao gồm diện tích phân hợp của predicted bounding box và ground truth.

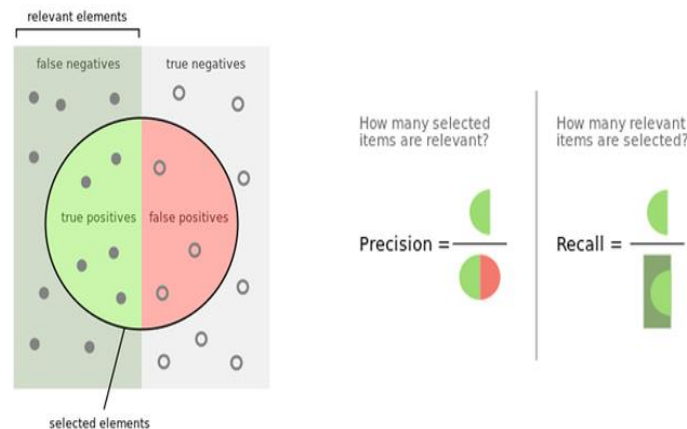
=> IOU càng lớn thì càng tốt, nhãn dự đoán càng gần với nhãn thực.

- Confusion Matrix:

		Model dự đoán	
		Positive	Negative
Thực tế	Positive	True Positive (Dự đoán Đúng là Positive)	False Negative (Dự đoán Sai là Negative)
	Negative	False Positive (Dự đoán Sai là Positive)	True Negative (Dự đoán Đúng là Negative)

Hình 17: Confusion Matrix^[6]

- Precision:



Hình 18: Precision^[6]

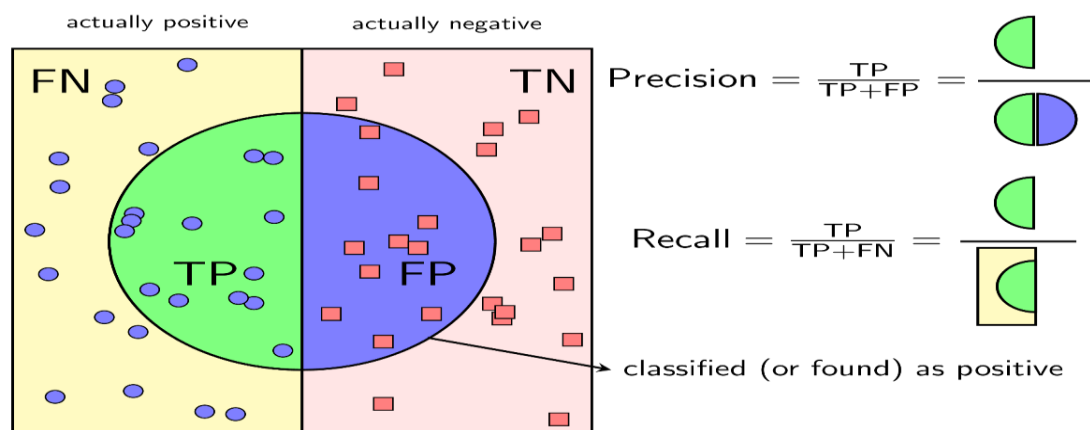
$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Hình 19: Công thức tính Precision^[6]

Precision là tỷ lệ giữa số sample được tính là True Positive (TP) với tổng số sample được phân loại là Positive (là tổng TP + FP).

=> Khi đó $0 < Precision \leq 1$, Precision càng lớn có nghĩa là độ chính xác của các điểm tìm được càng cao.

- Recall:



Hình 20: Recall^[7]

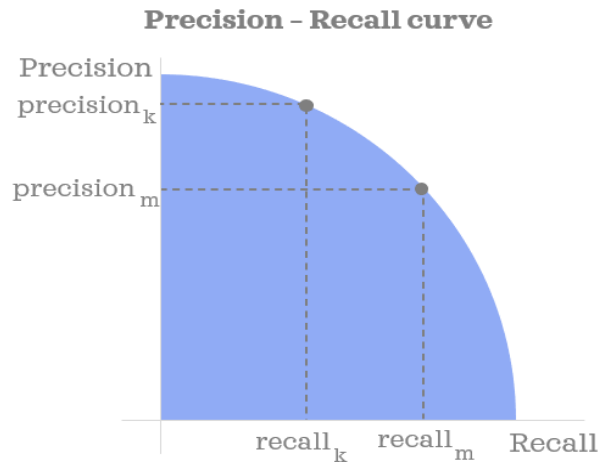
$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Hình 21: Công thức tính Recall^[7]

Recall là tỷ lệ giữa các điểm positive thực được nhận diện đúng trên tổng điểm positive thực.

=> Recall cao có nghĩa tỉ lệ bỏ sót các sample positive thực thấp.

- Precision Recall Curve:



Hình 22: Precision Recall Curve^[8]

Precision Recall Curve một cái nhìn chi tiết về khả năng của mô hình trong việc xác định các mẫu thuộc một lớp cụ thể và đánh giá sự cân đối giữa độ chính xác (precision) và độ phủ (recall).

- AP - Average Precision:

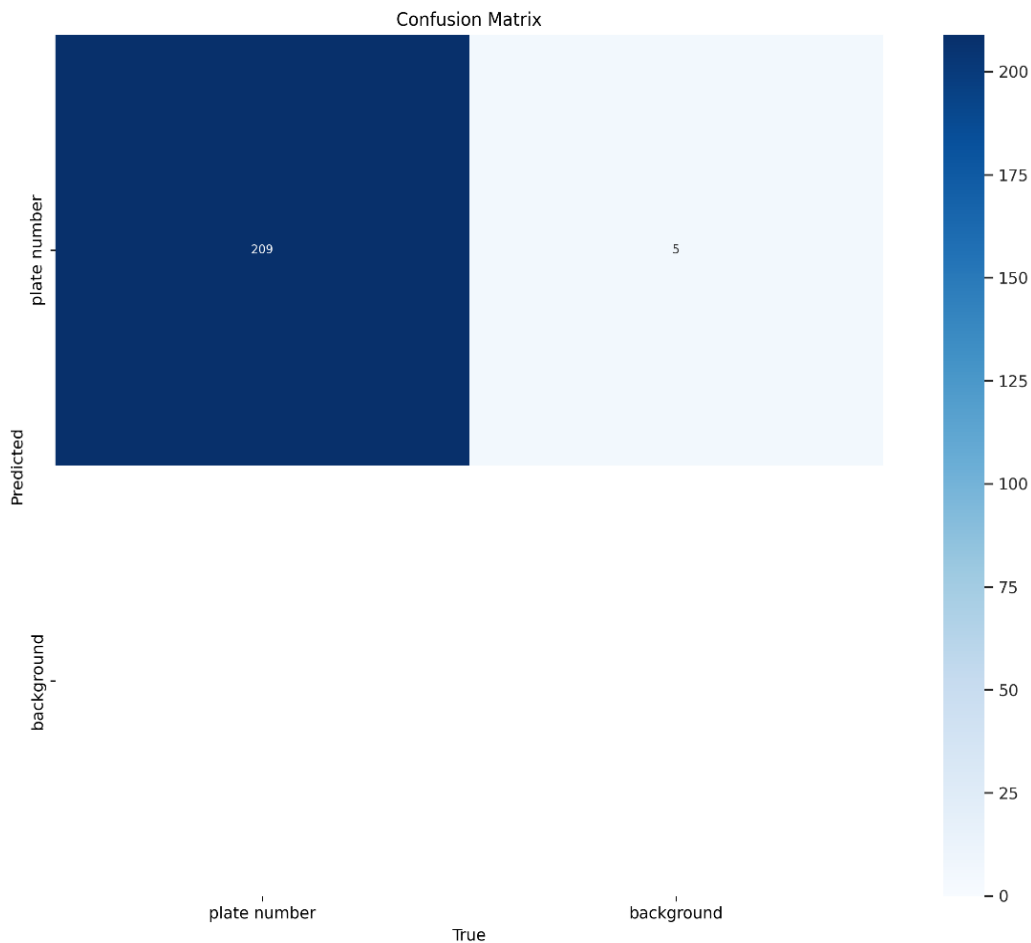
AP chính là vùng diện tích màu xanh nằm dưới đường cong Precision Recall.

- AP lớn nếu vùng này lớn, suy ra đường cong có xu hướng gần góc trên bên phải và có nghĩa là tại các threshold khác nhau thì Precision và Recall đều khá cao. Từ đó suy ra model tốt.
- AP nhỏ thì cả Precision và Recall đều khá thấp và model không tốt.

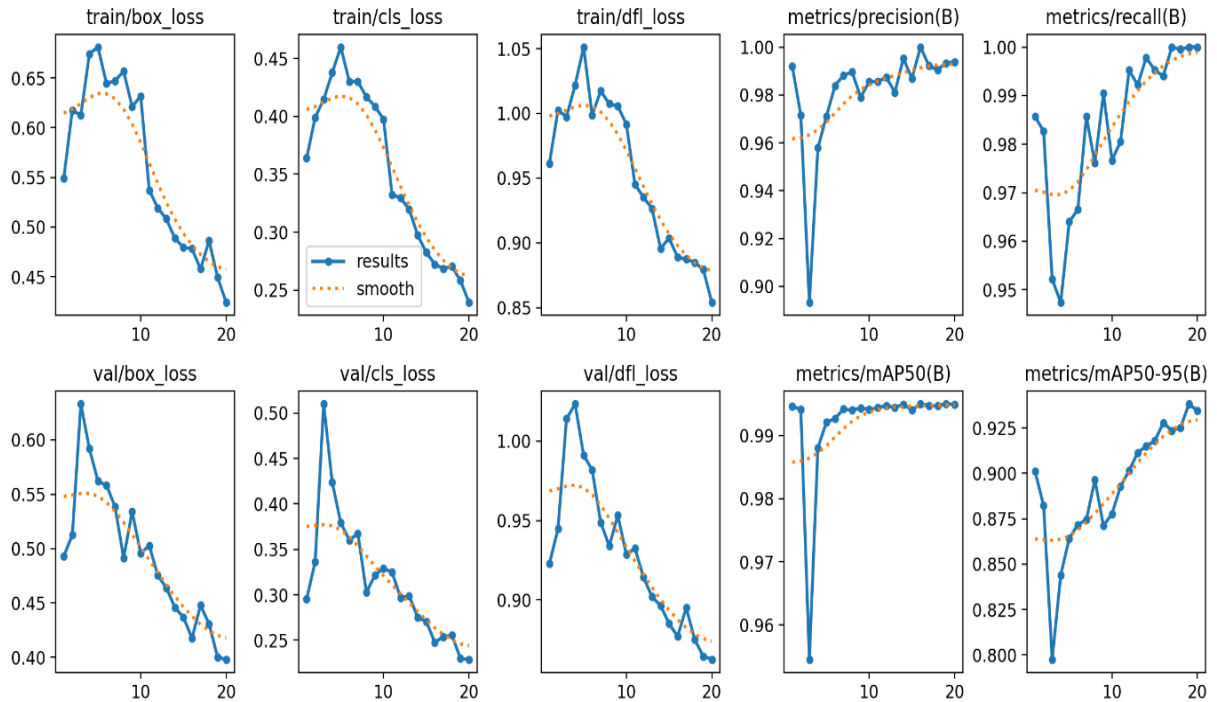
- mAP:
- Mean Average Precision là trung bình cộng giá trị AP của các class khác nhau.
- mAP50: đo lường độ chính xác của mô hình dựa trên một ngưỡng IoU (Intersection over Union) cố định là 0.5.
- mAP50-95: đo lường độ chính xác của mô hình trên một phạm vi rộng hơn của ngưỡng IoU từ 0.5 đến 0.95. Thay vì chỉ sử dụng ngưỡng IoU 0.5, mAP50-95 tính toán Precision-Recall trên các ngưỡng IoU từ 0.5 đến 0.95 và tính toán giá trị trung bình của chúng.

3.3. Kết quả phát hiện và nhận diện biển số xe theo mô hình YOLOv8

3.3.1. Kết quả mô hình



Hình 23: Confusion Matrix của mô hình phát hiện và nhận diện biển số xe



Hình 24: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 20 vòng cuối

Bảng 4: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 20 vòng cuối

Precision	0.99
Recall	0.99
mAP50	0.995
mAP50-95	0.937

Nhận xét: Các chỉ số precision, recall, mAP50, mAP50-95 rất cao hơn 0.9, cho thấy độ chính xác của khả năng phát hiện biển số xe (precision) rất cao, và độ nhạy (recall) cũng rất cao. Trung bình công hàm loss rất thấp cho thấy độ chính xác của việc dự đoán vị trí, loại và sự tồn tại của đối tượng trong các box là khá tốt.








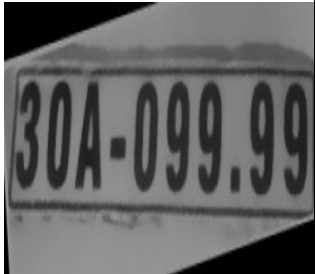

3.3.2. Kết quả thực tế

Bảng 5: Kết quả mô hình phát hiện và nhận diện biển số

Ảnh gốc	Ảnh dự đoán	Thời gian
		0.237s
		0.22s
		0.108s

3.4. Kết quả xoay ảnh biển số xe và sắp xếp ký tự

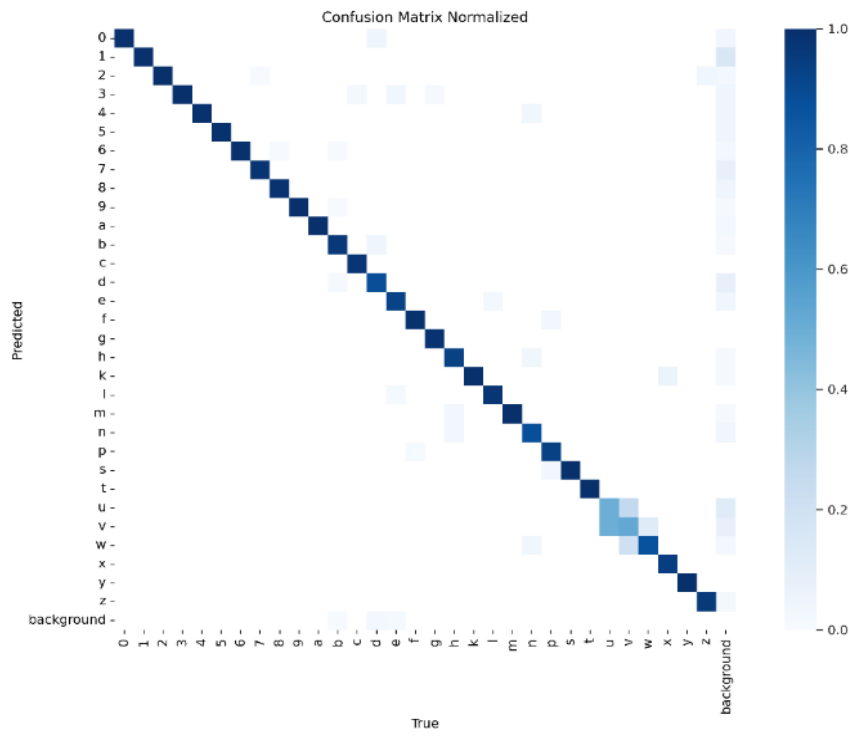
Bảng 6: Kết quả xoay ảnh biển số xe và sắp xếp ký tự

Ảnh gốc	Ảnh đã xoay	Ảnh ký tự	Góc lệch thực tế	Góc lệch sau khi xoay
			22°	1°
			18°	2°
			11°	1°

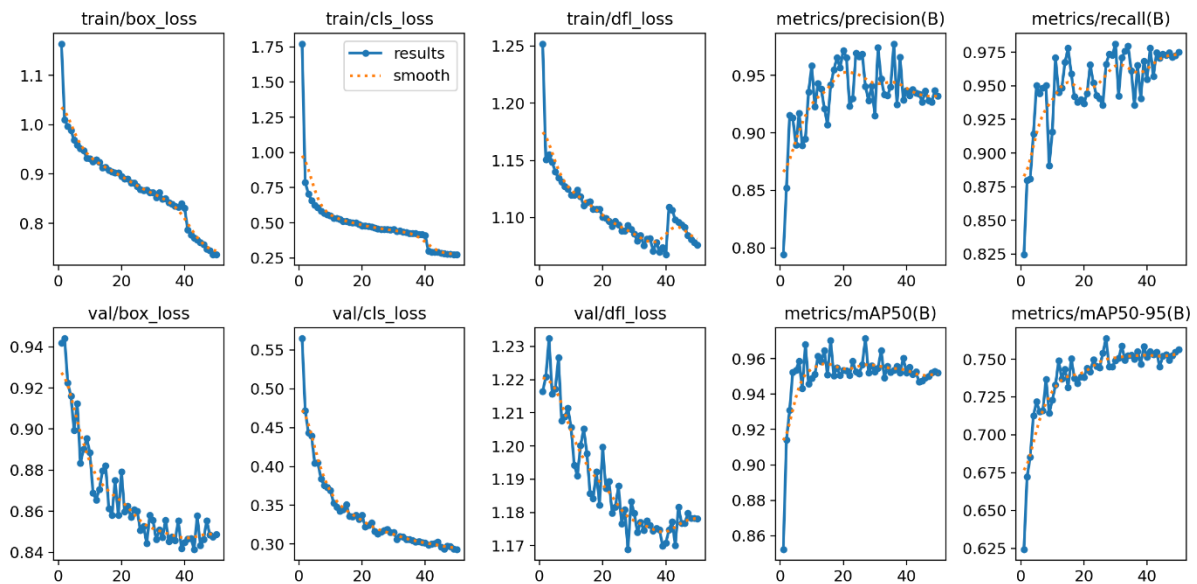
Nhận xét: Kết quả thuật toán xoay ảnh khá chính xác, căn chỉnh biển số theo phương ngang, từ đó cải thiện quá trình trích xuất thông tin và quản lý dữ liệu biển số xe một cách chính xác và hiệu quả.

3.5. Kết quả phát hiện và nhận diện ký tự theo mô hình YOLOv8

3.5.1. Kết quả mô hình



Hình 25: Confusion Matrix Normalized của mô hình phát hiện và nhận diện ký tự



Hình 26: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 50 vòng cuối

Bảng 7: Kết quả của hàm loss, precision, recall, mAP50 và mAP50-95 sau khi train 50 vòng cuối

Precision	0.96
Recall	0.96
mAP50	0.952
mAP50-95	0.756

Nhân xét: Các chỉ số precision, recall, mAP50, mAP50-95 rất cao (hơn 0.9), cho thấy độ chính xác của khả năng phát hiện biên số xe (precision) rất cao, và độ nhạy (recall) cũng rất cao. Trung bình công hàm loss rất thấp cho thấy độ chính xác của việc dự đoán vị trí, loại và sự tồn tại của đối tượng trong các box là khá tốt.

3.5.2. Kết quả thực tế

Bảng 8: Kết quả phát hiện và nhận diện ký tự

Ảnh đầu vào	Kết quả nhận dạng	Thời gian nhận dạng
		0.2s
		0.15s
		0.21s

3.6. Công nghệ và framework đã sử dụng

Django^[3]: Framework phát triển ứng dụng web bằng Python, cung cấp các công cụ và khung công việc để xây dựng ứng dụng web.

SQLite^[10]: Là một hệ thống quản lý cơ sở dữ liệu quan hệ nhẹ (lightweight), thường được tích hợp sẵn và sử dụng như một phần mềm cơ sở dữ liệu mặc định trong Django cho môi trường phát triển và triển khai ứng dụng web đơn giản.

3.7. Kết quả API

Module Server cung cấp 7 APIs với các chức năng chính như đóng-mở servo, lấy tín hiệu xe ra-vào, hiển thị thông tin lên màn hình LCD.

Web Server cung cấp 14 APIs cho người dùng và admin quản lý bãi đỗ.

3.8. Ứng dụng phần mềm

Hình 27: Giao diện ứng dụng phần mềm

3.9. Ứng dụng web

- ✓ Hoàn thành thiết kế giao diện:

Giao diện đăng nhập.

Giao diện đăng ký tài khoản.

Giao diện trang chủ.

Giao diện quản lý người dùng.

Giao diện xem lịch sử đỗ xe.

Giao diện thông tin cá nhân người dùng.

Giao diện thông tin lịch sử đỗ xe của người dùng.

✓ Hoàn thành các chức năng chính:

Đăng nhập, đăng ký.

Lưu thông tin xe vào, ra bãi đỗ.

Xem, sửa, xóa thông tin người dùng.

Xem lịch sử đỗ xe.

Thay đổi thông tin cá nhân người dùng.

✓ Giao diện web:

- Giao diện thông tin người dùng

Cập nhật thông tin

Xem lịch sử gửi xe

Đăng xuất

Họ và tên: Nguyễn Việt Hoài Bảo

Số điện thoại: 0786441442

Địa chỉ: Đà Nẵng

Cập nhật thông tin

Thêm biển số

Số thứ tự	Biển số xe
1	67A01703

Hình 28: Giao diện thông tin người dùng

- Giao diện lịch sử gửi xe của người dùng

HỆ THỐNG QUẢN LÝ BÃI ĐỖ XE				
Cập nhật thông tin Xem lịch sử gửi xe Đăng xuất	LỊCH SỬ GỬI XE			
	Biển số xe	Thời Gian Vào	Thời Gian Ra	Giá tiền
	67A01703	08-12-2023 : 11:37 a.m.	08-12-2023 : 11:39 a.m.	10000 VND
	67A01703	08-12-2023 : 11:43 a.m.	08-12-2023 : 11:44 a.m.	10000 VND

Hình 29: Giao diện lịch sử gửi xe người dùng

- Giao diện quản lý tài khoản

HỆ THỐNG QUẢN LÝ BÃI ĐỖ XE						
Trang chủ Quản lý tài khoản Xem lịch sử gửi xe Đăng xuất	<input type="text" value="Nhập biển số xe"/> <input type="button" value="Q"/> <input type="text" value="Nhập tên"/> <input type="button" value="Q"/> <input type="button" value="Thêm tài khoản"/>					
	Họ và Tên	Số điện thoại	Địa chỉ	Biển số xe		
	Nguyễn Văn Nam	1234560101	Quảng Nam	• 29B100111	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>
	Nguyễn Văn Hiếu	1234561234	Đà Nẵng	• 43H132277	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>
	Phùng Thanh Tùng	0905873002	Hà Nội	• 92H101234	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>
	Nguyễn Việt Hoài Bảo	0786441442	Đà Nẵng	• 67A01703	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>
	Nguyễn Đức Dũng	1234561234	Tỉnh Thanh Hóa	• 30A09999	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>
	Nguyễn Phạm Nhật Vỹ	0905874211	Đà Nẵng	• 43H131166	<input type="button" value="Chỉnh sửa"/>	<input type="button" value="Xóa"/>

Hình 30: Giao diện quản lý tài khoản

- Giao diện quản lý lịch sử gửi xe

HỆ THỐNG QUẢN LÝ BÃI ĐỖ XE					
Trang chủ Quản lý tài khoản Xem lịch sử gửi xe Đăng xuất	LỊCH SỬ GỬI XE				
	Biển số xe	Thời Gian Vào	Thời Gian Ra	Giá tiền	
	30A09999	11-11-2023 : 10:50 p.m.	12-11-2023 : 1:20 a.m.	24900 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	29B100111	08-11-2023 : 11:52 p.m.	11-11-2023 : 10:33 p.m.	434100 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	92H101234	07-11-2023 : 4:15 p.m.	07-11-2023 : 9:34 p.m.	41800 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	30A09999	20-11-2023 : 8:06 p.m.	23-11-2023 : 11:25 p.m.	461900 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	43H132277	07-11-2023 : 9:57 p.m.	08-11-2023 : 9:16 p.m.	149800 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	67A01703	08-12-2023 : 11:37 a.m.	08-12-2023 : 11:39 a.m.	10000 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	67A01703	08-12-2023 : 11:43 a.m.	08-12-2023 : 11:44 a.m.	10000 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>
	30F25658	08-12-2023 : 11:37 a.m.	08-12-2023 : 11:38 a.m.	10000 VND	<input type="button" value="Chi tiết"/> <input type="button" value="Xóa"/>

Hình 31: Giao diện quản lý lịch sử gửi xe

4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Thiết kế:

Hệ thống được thiết kế gọn gàng với trọng lượng nhẹ, tạo điều kiện thuận lợi cho di chuyển linh hoạt và tiện ích. Đồng thời, khả năng mở rộng của nó không chỉ giới hạn ở cấu hình hiện tại mà còn mở ra tiềm năng phát triển với cả phần cứng và phần mềm trong tương lai.

Chức năng:

Phát hiện biển số: Hệ thống đã thực hiện tốt trong việc phát hiện biển số xe với độ chính xác lên đến 93.7%, phát hiện hầu hết các dạng biển số và tốc độ xử lý ổn định.

Nhận diện biển số: Mô hình nhận diện biển số xe đạt kết quả cao với hiệu suất lên đến 95.2% và tốc độ xử lý ổn định khoảng 0.6s. Tuy nhiên, độ chính xác thực tế có thể thấp hơn vì ảnh hưởng từ độ phân giải camera và điều kiện thực tế như môi trường, ánh sáng.

Ứng dụng web: Thực hiện tốt và đầy đủ các chức năng cơ bản của một ứng dụng quản lý bãi đỗ xe. Giao diện ở mức ổn, dễ sử dụng, đáp ứng được nhu cầu của doanh nghiệp và người dùng.

4.2. Hướng phát triển

Để có thể cải thiện được độ chính xác và tốc độ xử lý của hệ thống, cần có hướng phát triển và triển khai trong tương lai:

- Phần cứng: Sử dụng camera tiên tiến, có độ phân giải cao trong việc xử lý hình ảnh
- Thuật toán phát hiện biên số và nhận diện biên số: Tìm kiếm thêm nhiều mô hình mới, cải thiện độ chính xác và tốc độ của thuật toán.
- Ứng dụng web: Thiết kế giao diện tốt hơn và thân thiện với người dùng, phát triển thêm nhiều tính năng chi tiết, giúp quản lý và bảo vệ thông tin cá nhân người dùng. Bên cạnh đó, cải thiện server để đáp ứng số lượng truy cập lớn.

5. DANH MỤC TÀI LIỆU THAM KHẢO

- [1] RESTful API, <https://topdev.vn/blog/restful-api-la-gi/>
- [2] Mô hình YOLO, [Mô Hình Nhận Dạng Vật Thể YOLOv8 - HackMD](#)
- [3] Framework Django, <https://docs.djangoproject.com/en/4.2/>
- [4] Mean Average Precision, [Mean Average Precision \(mAP\): A Complete Guide \(kili-technology.com\)](#)
- [5] Công thức IoU, [Mean Average Precision \(mAP\) Explained: Everything You Need to Know \(v7labs.com\)](#)
- [6] Confusion Matrix, [Đánh giá model AI với Precision, Recall và F1 Score - Mì AI \(miai.vn\)](#)
- [7] Recall, [Machine Learning cơ bản \(machinelearningcoban.com\)](#)
- [8] Precision Recall Curve, [SuNT's Blog | AI in Practical \(tiensu.github.io\)](#)
- [9] Thuật toán xoay ảnh, [Nhận diện và trích xuất thông tin biển số xe Việt Nam \(viblo.asia\)](#)
- [10] SQLite, <https://developer.android.com/training/data-storage/sqlite?hl=vi>