

NGUYỄN VĂN TUẤN

KHOA HỌC MÁY TÍNH

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH KHOA HỌC MÁY TÍNH

XÂY DỰNG MÔ HÌNH NHẬN DIỆN BIỂN SỐ XE  
SỬ DỤNG WPOD

CBHD: ThS. Nguyễn Thị Hương Lan

Sinh viên: Nguyễn Văn Tuấn

Mã số sinh viên: 2020601830

Hà Nội – Năm 2024

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  
**KHOA CÔNG NGHỆ THÔNG TIN**

-----

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**XÂY DỰNG MÔ HÌNH NHẬN DIỆN BIỂN SỐ XE SỬ DỤNG WPOD**

**CBHD : Ths. Nguyễn Thị Hương Lan**  
**Sinh viên : Nguyễn Văn Tuấn**  
**Mã sinh viên : 2020601830**

**Hà Nội, Năm 2024**

## LỜI MỞ ĐẦU

Công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lý được các công việc một cách tự động, phục vụ cho lợi ích của con người. Kèm theo đó là sự bùng nổ của các thiết bị ghi hình như: điện thoại di động, máy tính bảng, máy ảnh, các hệ thống camera giám sát (CCTV) ,... Trong những năm gần đây, nhiều tiến bộ trong lĩnh vực nghiên cứu trí tuệ nhân tạo, học máy và học sâu đã đưa ra những giải pháp cho các bài toán về nhận dạng như nhận dạng vật thể, nhận diện kí tự, nhận dạng khuôn mặt, nhận dạng hình dáng, nhận dạng giọng nói,... với độ chính xác rất cao. Những công nghệ trên hoàn toàn có khả năng ứng dụng vào giải quyết các vấn đề liên quan đến các phương tiện đi lại và nhận diện biển số xe là một trong số đó.

Nhận diện biển số xe là một bài toán được ứng dụng rất rộng rãi trong các công việc trong xã hội. Một trong những ứng dụng tiêu biểu của nhận diện biển số xe là trong việc quản lý các phương tiện trong các bãi đỗ xe ở trong chung cư, siêu thị, trường đại học, v. v... Bên cạnh đó, nó còn có thể áp dụng trong việc hỗ trợ những vấn đề liên quan đến giao thông như: trạm thu phí không dừng ở trên những cao tốc hay việc hỗ trợ trích xuất biển số xe cho những phương tiện vi phạm luật giao thông.

Dù là một bài toán cơ bản và rất đơn giản nhưng nó cũng tồn tại một số thách thức liên quan đến việc nhận diện kí tự trong biển số xe như: vì biển số xe có kích thước khá nhỏ nên khi trích xuất biển số xe ra thì chất lượng hình ảnh của biển rất tệ; hay là việc khi ta dùng các mô hình xử lý việc phát hiện biển số như yolo thì đầu ra ta thu được là một hình chữ nhật có chứa biển số, nếu như biển số ở phía chính diện thì không sao nhưng nếu bị lệch thì sẽ chứa một phần cảnh quan điều đó sẽ làm giảm đi độ chính xác của nhận diện kí tự của biển số,... Vì vậy, để giải quyết vấn đề bị lệch của biển số xe tôi quyết định lựa chọn đề tài “Xây dựng mô hình nhận diện biển số xe sử dụng WPOD”.

Để có thể thực hiện đề tài, đầu tiên, tôi sẽ tiến hành nghiên cứu và phân tích bài toán nhận diện biển số xe. Sau đó, tôi tìm kiếm và tham khảo các nghiên cứu liên quan cần thiết có thể hỗ trợ giải quyết bài toán. Quá trình thu thập và tiền xử lý dữ liệu sẽ được thực hiện để có một bộ dữ liệu đầy đủ phục vụ quá trình huấn luyện và đánh giá các mô hình. Tiếp theo, tôi tiến hành thực nghiệm kết quả và xây dựng chương trình thử nghiệm.

Đối tượng nghiên cứu của đề tài bao gồm công nghệ nhận diện biển số xe tự động (ALPR) . Phạm vi nghiên cứu được giới hạn trong phân tích, đánh giá hiệu suất của công nghệ nhận diện biển số xe hiện có trong việc phát hiện và nhận diện các kí tự trong biển số xe.

Bố cục đề tài bao gồm bốn chương chính:

Chương 1. Tổng quan về bài toán nhận dạng biển số xe: Giới thiệu tổng quan về xử lý ảnh, thị giác máy tính, công nghệ nhận diện biển số xe, lịch sử phát triển, ứng dụng trong thực tiễn, và khó khăn thách thức.

Chương 2. Cơ sở lý thuyết: tổng hợp các nghiên cứu có liên quan về phương pháp giải quyết từng bài toán con trong đề tài và đưa ra phương pháp chính sẽ sử dụng trong báo cáo.

Chương 3. Xây dựng và cài đặt bài toán: trình bày công cụ, quy trình xây dựng, phân tích thiết kế phần mềm demo.

Chương 4. Thực nghiệm: kết quả khi áp dụng vào thực tế

Qua đề tài này, tôi hy vọng đề tài sẽ có thể giúp ích cho sự phát triển công nghệ nhận diện biển số xe ở Việt Nam.

## MỤC LỤC

<b>LỜI MỞ ĐẦU.....</b>	<b><i>i</i></b>
<b>MỤC LỤC.....</b>	<b><i>iii</i></b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b><i>v</i></b>
<b>DANH MỤC BẢNG BIỂU.....</b>	<b><i>vi</i></b>
<b>DANH MỤC NHỮNG TỪ VIẾT TẮT.....</b>	<b><i>vii</i></b>
<b>LỜI CẢM ƠN.....</b>	<b><i>viii</i></b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....</b>	<b><i>1</i></b>
<b>1.1. Tổng quan về xử lý ảnh .....</b>	<b><i>1</i></b>
1.1.1. Khái niệm về ảnh .....	<i>1</i>
1.1.2. Khái niệm về điểm ảnh .....	<i>2</i>
1.1.3. Xử lý ảnh.....	<i>2</i>
1.1.4. Các vấn đề cơ bản của xử lý ảnh .....	<i>3</i>
1.1.5. Các phép xử lý ảnh cơ bản.....	<i>7</i>
1.1.6. Ứng dụng của xử lý ảnh.....	<i>8</i>
<b>1.2. Tổng quan về thị giác máy tính .....</b>	<b><i>8</i></b>
1.2.1. Khái niệm .....	<i>8</i>
1.2.2. Ứng dụng trong thực tiễn .....	<i>9</i>
1.2.3. Các lĩnh vực liên quan .....	<i>10</i>
<b>1.3. Tổng quan về công nghệ nhận dạng biển số xe .....</b>	<b><i>11</i></b>
1.3.1. Lịch sử phát triển .....	<i>11</i>
1.3.2. Khó khăn và thách thức .....	<i>12</i>
<b>1.4. Tổng quan về đề tài.....</b>	<b><i>13</i></b>
1.4.1. Tên đề tài.....	<i>13</i>
1.4.2. Lý do chọn đề tài.....	<i>13</i>
1.4.3. Mục tiêu của đề tài .....	<i>14</i>
1.4.4. Đối tượng và phạm vi .....	<i>15</i>
1.4.5. Kết quả dự kiến đạt được .....	<i>15</i>
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....</b>	<b><i>16</i></b>
<b>2.1. Bài toán nhận diện vật thể.....</b>	<b><i>16</i></b>
2.1.1. Các phương pháp .....	<i>16</i>
2.1.2. Phương pháp đánh giá mô hình .....	<i>17</i>

2.2.	Lịch sử phát triển YOLO .....	19
2.3.	Nhận dạng biển số xe .....	27
<b>CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....</b>		<b>29</b>
3.1.	Tổng quan về hệ thống.....	29
3.2.	Xây dựng hệ thống .....	29
3.2.1.	Xác định vùng chứa biển số xe sử dụng YOLOv9 .....	30
3.2.2.	Xác định vùng chứa biển số sử dụng WPOD-NET .....	36
3.2.3.	Phân đoạn từng ký tự trên biển số.....	40
3.2.4.	Phân loại ký tự sử dụng mô hình CNN.....	42
3.2.5.	Định dạng lại biển số .....	43
<b>CHƯƠNG 4: THỰC NGHIỆM.....</b>		<b>44</b>
4.1.	Chuẩn bị bộ dữ liệu.....	44
4.1.1.	Bộ dữ liệu phát hiện biển số xe.....	44
4.1.2.	Bộ dữ liệu nhận diện biển số xe .....	44
4.2.	Huấn luyện mô hình nhận diện vật thể.....	45
4.3.	Đánh giá trên tập thực nghiệm .....	46
4.3.1.	Mô hình phát hiện biển số xe .....	46
4.3.2.	Mô hình nhận dạng biển số xe .....	47
4.4.	Triển khai hệ thống.....	48
<b>KẾT LUẬN.....</b>		<b>49</b>
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>50</b>

## DANH MỤC HÌNH ẢNH

Hình 1.1 Xử lý ảnh.....	2
Hình 1.2 Các giai đoạn của quá trình xử lý ảnh.....	3
Hình 1.3 Ảnh nhiễu và sau khi lọc nhiễu.....	4
Hình 1.4 Thị giác máy tính .....	10
Hình 1.5 Kết quả trả về khi sử dụng YOLO để phát hiện biển số .....	14
Hình 2.1 Ma trận hỗn loạn .....	18
Hình 2.2 Intersection of union .....	19
Hình 2.3 Kiến trúc mạng YOLOv1.....	21
Hình 2.4 Kiến trúc mạng YOLOv3.....	22
Hình 2.5 Kiến trúc mạng YOLOv8.....	26
Hình 3.1 Tổng quan về hệ thống.....	29
Hình 3.2 Kết quả thực nghiệm với $k_x = k_y = 0.5$ .....	35
Hình 3.3 Kết quả thực nghiệm với $k_x = k_y = 1.9$ .....	35
Hình 3.4 Kết quả thực nghiệm với $k_x = k_y = 2.9$ .....	35
Hình 3.5 Kết quả thực nghiệm với $k_x = 1.7$ và $k_y = 2.9$ .....	36
Hình 3.6 Hoạt động của mạng WPOD.....	37
Hình 3.7 Kiến trúc mạng WPOD .....	38
Hình 3.8 Kết quả sau khi sử dụng WPOD-NET .....	40
Hình 3.9 Ví dụ về kết quả sau khi áp dụng adaptive threshold .....	41
Hình 3.10 Kết quả sau khi phân đoạn từng kí tự .....	42
Hình 4.1 Ảnh ví dụ tại điểm khảo sát .....	44
Hình 4.2 Một vài hình ảnh trong bộ dữ liệu của công ty GreenParking. ....	45
Hình 4.3 Kết quả huấn luyện mô hình phát hiện vật thể .....	46
Hình 4.4 Một số hình ảnh kết quả của mô hình phát hiện biển số xe.....	46
Hình 4.5 Một số hình ảnh kết quả của mô hình nhận dạng biển số xe ...	47
Hình 4.6 Một số hình ảnh kết quả của việc triển khai mô hình.....	48

**DANH MỤC BẢNG BIỂU**

Bảng 1 Tác dụng của xử lý ảnh trong các lĩnh vực áp dụng.....	8
Bảng 2 Hiệu suất của YOLOv9 trên tập dữ liệu MS COCO .....	31
Bảng 3 Kết quả thực nghiệm xác định hằng số mở rộng $k_x$ và $k_y$ .....	33
Bảng 4 Kết quả nhận dạng biển số xe trên bộ dữ liệu GreenParking .....	47



**DANH MỤC NHỮNG TỪ VIẾT TẮT**

OCR	Optical Character Recognition
CCTV	Closed Circuit Television
CNN	Convolutional Neural Networks
GPU	Graphics Processing Unit
YOLO	You Only Look Once
ALPR	Automatic License Plate Recognition
WPOD	Wrap Planer Object Detection Network

## LỜI CẢM ƠN

Trong quá trình thực hiện báo cáo của học phần đồ án tốt nghiệp, với đề tài "Xây dựng mô hình nhận diện biển số xe sử dụng WPOD", tôi xin gửi lời cảm ơn chân thành đến tất cả những người đã đóng góp và hỗ trợ hoàn thành thành công báo cáo này.

Đầu tiên, tôi muốn gửi lời cảm ơn sâu sắc tới giảng viên hướng dẫn, thạc sĩ Nguyễn Thị Hương Lan, vì sự chỉ dẫn, hướng dẫn và những kiến thức quý báu mà cô đã truyền đạt cho tôi trong suốt quá trình thực hiện báo cáo cũng như trong quá trình học. Sự kiên nhẫn và tận tâm của cô đã giúp tôi vượt qua những khó khăn và hoàn thiện báo cáo một cách tốt nhất.

Tôi cũng muốn bày tỏ lòng biết ơn đến khoa Công nghệ thông tin trường Đại học Công Nghiệp Hà Nội đã tạo điều kiện thuận lợi và cung cấp những kiến thức chuyên môn quan trọng cho tôi trong suốt quá trình học tập và nghiên cứu tại trường.

Tôi không thể không nhắc đến sự giúp đỡ của các bạn bè cùng lớp trong việc tìm kiếm thông tin, trao đổi ý kiến và hỗ trợ kỹ thuật.

Tôi cũng muốn bày tỏ lòng biết ơn đến gia đình và người thân đã luôn ủng hộ, động viên và hiểu rõ những khó khăn tôi gặp phải trong quá trình thực hiện báo cáo này. Sự động viên và tình yêu thương của gia đình là nguồn động lực quan trọng giúp tôi vượt qua mọi khó khăn.

Cuối cùng, tôi xin gửi lời cảm ơn chân thành đến tất cả những người đã đọc và đánh giá báo cáo này. Sự quan tâm và góp ý của mọi người là động lực để tôi tiếp tục nỗ lực và hoàn thiện hơn trong những nghiên cứu và dự án tương lai.

Xin chân thành cảm ơn!

Nguyễn Văn Tuấn

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Tổng quan về xử lý ảnh

Xử lý ảnh là một lĩnh vực quan trọng trong trí tuệ nhân tạo và học máy, là lĩnh vực nghiên cứu về cách xử lý và phân tích hình ảnh số để trích xuất ra những thông tin hữu ích từ hình ảnh.

### 1.1.1. Khái niệm về ảnh

Ảnh là thông tin về vật thể hay quang cảnh được chiếu sáng mà con người quan sát được và cảm nhận được bằng mắt và hệ thống thần kinh thị giác. Ảnh được tạo thành từ ba yếu tố:

- Vật thể, không gian quan sát được chiếu sáng.
- Nguồn sáng.
- Cảm nhận (mắt) .

Ảnh trong tự nhiên (ảnh liên tục) là những tín hiệu liên tục về không gian và giá trị độ sáng. Tín hiệu thuộc loại tín hiệu đa chiều: tọa độ  $(x,y,z)$  , độ sáng  $(\lambda)$  , thời gian  $(t)$  .

Ảnh được lưu trữ trong máy tính như một mảng 2 chiều chứa giá trị số. Các số tương ứng với các thông tin khác nhau như màu sắc hay cường độ mức xám, độ chói, thành phần màu... Để có thể lưu trữ và biểu diễn ảnh trên máy tính (ảnh số) con người phải tiến hành biến đổi các tín hiệu liên tục đó thành một số hữu hạn các tín hiệu rời rạc thông qua quá trình lượng tử hóa và lấy mẫu thành phần giá trị độ sáng. Ảnh trong không gian 2 chiều được định nghĩa là một hàm 2 biến  $S(x,y)$  , với  $S$  là giá trị độ sáng tại tọa độ  $(x,y)$  .

Với ảnh liên tục  $S(x,y)$ : Miền xác định  $(x,y)$  liên tục, miền giá trị  $S$  liên tục.

Với ảnh số  $S(m,n)$ : Là ảnh liên tục được số hóa, miền xác định  $(m,n)$  rời rạc, miền giá trị  $S$  rời rạc.

Ảnh số: Tín hiệu số 2D.

### 1.1.2. Khái niệm về điểm ảnh

Điểm ảnh được xem là dấu hiệu hay cường độ sáng tại một tọa độ trong không gian của đối tượng. Ảnh được xem như một tập hợp các điểm ảnh được tổ chức dưới dạng ma trận các điểm ảnh gồm M dòng và N cột. Giao giữa dòng và cột là điểm ảnh (pixel) . Mỗi điểm ảnh gồm hai thông số: tọa độ (x,y) và giá trị màu/mức xám.

Màu (color): Là giá trị được tổ hợp từ ba thành phần cơ bản: Đỏ (Red) , Xanh lam (Blue) , Xanh lục (Green) . Trong không gian màu RGB, một màu được tổ hợp từ ba thành phần R, G, B theo công thức  $\text{color} = R + G \cdot 2^8 + B \cdot 2^{16}$ . Nếu ba thành phần này không đồng thời bằng nhau thì giá trị tổ hợp trên sẽ có sắc thái màu, khi đó ta gọi là ảnh màu.

Mức xám (Gray level): Màu xám là màu có ba thành phần  $R = G = B$ . Ảnh xám hay ảnh đa cấp xám là ảnh chỉ chứa toàn màu xám.

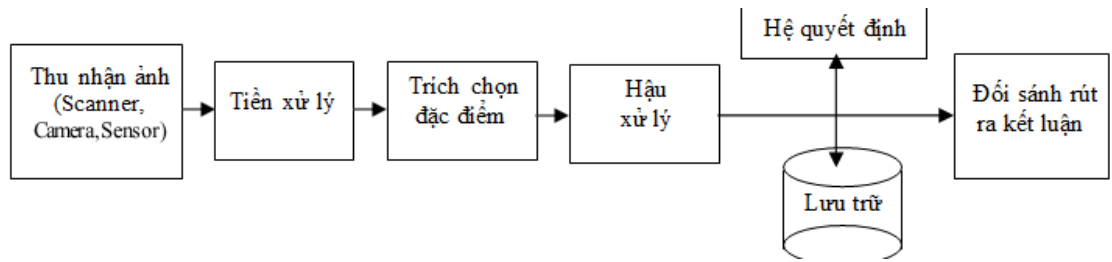
### 1.1.3. Xử lý ảnh

Quá trình xử lý ảnh gồm một dãy các thao tác trên ảnh đầu vào nhằm cho ra kết quả mong muốn. Đầu ra của quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1.1 Xử lý ảnh

Xử lý ảnh là một tiến trình gồm nhiều công đoạn nhỏ, các công đoạn đó bao gồm:



Hình 1.2 Các giai đoạn của quá trình xử lý ảnh

- Thu nhận ảnh: Việc thu nhận có thể thực hiện qua các thiết bị như máy ảnh, ảnh chụp từ vệ tinh qua các bộ cảm biến (Sensors) , qua máy quét (Scanners) .
- Tiền xử lý: Nhằm nâng cao chất lượng ảnh đầu vào để làm nổi bật một số đặc điểm của ảnh đầu vào hay làm cho ảnh giống nhất với trạng thái gốc của nó. Có nhiều công cụ khác nhau để thực hiện tùy thuộc vào trạng thái của ảnh đầu vào như:
  - Xóa nhiễu: Loại bỏ các đối tượng dư thừa trong ảnh (có thể đo chất lượng của thiết bị thu nhận, do nguồn sáng) .
  - Nắn chỉnh hình học: Khắc phục các biến dạng do các thiết bị điện tử và quang học gây nên, có thể khắc phục bằng các phép chiếu.
  - Chỉnh mức xám: Khắc phục tính không đồng đều của mức xám, thường dùng để tăng hoặc giảm số mức xám của ảnh.
- Trích chọn đặc trưng: Nhằm tóm tắt ảnh. Có thể sử dụng các công cụ như: Dò biên để xác định biên, phân vùng, làm mảnh để trích xương,...
- Hậu xử lý: Nhằm hiệu chỉnh lại đặc điểm của những đặc trưng đã trích được từ bước bên trên sao cho bước thực hiện tiếp theo được thuận tiện và nhanh chóng nhưng vẫn không làm ảnh hưởng đến kết quả.
- Tùy mục đích của ứng dụng mà chuyển sang giai đoạn khác là lưu trữ, nhận dạng, phân lớp để rút ra kết luận, ...

#### 1.1.4. Các vấn đề cơ bản của xử lý ảnh

##### a. Khử nhiễu

Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh:

- Nhiễu hệ thống: Là nhiễu có quy luật có thể khử bằng các phép biến đổi trên miền tần số.
- Nhiễu ngẫu nhiên: Là dạng nhiễu xuất hiện ngẫu nhiên và có thể theo một phân bố nào đó. Chúng có thể được loại bỏ bằng các phép lọc.



Hình 1.3 Ảnh nhiễu và sau khi lọc nhiễu

#### b. *Chỉnh mức xám*

Chỉnh số mức xám khắc phục tính không đồng đều mức xám do hệ thống gây ra. Thông thường có 2 hướng tiếp cận:

- Giảm số mức xám: Thực hiện bằng cách nhón các mức xám gần nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng là in ảnh màu ra máy in đen trắng.
- Tăng mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh.

#### c. *Phân tích ảnh*

Là khâu quan trọng trong quá trình xử lý ảnh để tiến tới hiểu ảnh. Trong phân tích ảnh, việc trích chọn đặc điểm là một bước quan trọng. Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau:

- Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biến độ điểm uốn,...
- Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering) . Các bộ vùng được gọi là mặt nạ đặc điểm (feature mask) thường là các khe hẹp với hình ảnh khác nhau (chữ nhật, tam giác, cung tròn ...)
- Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng rất hữu ích trong việc trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử Laplace, ...

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng lưu trữ nhỏ.

#### *d. Nhận dạng*

Nhận dạng tự động (automatic recognition) , mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể sử dụng các phương pháp sau:

- Phân loại có mẫu (phân lớp): Trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.
- Phân loại không có mẫu (phân cụm): Trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau:

- Thu nhận dữ liệu và tiền xử lý.
- Biểu diễn dữ liệu.
- Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng:

- Đối sánh mẫu dựa trên các đặc trưng được chính chọn.
- Phân loại thống kê.
- Đối sánh cấu trúc.
- Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng, không thể chỉ dùng cho một cách tiếp cận đơn lẻ để phân loại “tối ưu”. Thay vào đó, ta cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới này sinh trong cuộc sống không chỉ tạo ra những thách thức về mặt giải thuật, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả các ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

#### *e. Nén ảnh*

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả 2 khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở 2 khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

- Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần suất xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà chiến lược mã hóa thích hợp.
- Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa. Cụ thể là lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau để tiến hành mã hóa.
- Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn dựa trên các phép biến đổi trên miền tần số.
- Nén ảnh Fractal: Sử dụng tính chất Fractal của các đối tượng ảnh, thể hiện sự lặp lại của các chi tiết. Kỹ thuật nén sẽ tính toán để chỉ



cần lưu trữ phần gốc ảnh và quy luật sinh ra ảnh theo nguyên lý Fractal.

### 1.1.5. Các phép xử lý ảnh cơ bản

Có nhiều phép xử lý ảnh được sử dụng trong thị giác máy tính để trích xuất thông tin và phân tích hình ảnh. Dưới đây là một số phép xử lý ảnh phổ biến:

*Chuyển đổi không gian màu:* Chuyển đổi không gian màu giúp chúng ta chuyển đổi hình ảnh từ không gian màu này sang không gian màu khác, ví dụ như chuyển đổi từ RGB sang Grayscale. Điều này giúp chúng ta giảm kích thước dữ liệu và tăng tốc độ xử lý.

*Làm mờ (Blurring):* Làm mờ là một phép xử lý ảnh được sử dụng để giảm nhiễu và loại bỏ chi tiết không cần thiết trong hình ảnh. Các phương pháp làm mờ phổ biến bao gồm làm mờ Gaussian và làm mờ trung bình.

*Phát hiện biên:* Phát hiện biên là một phép xử lý ảnh được sử dụng để tìm ra các đường biên trong hình ảnh. Các phương pháp phát hiện biên phổ biến bao gồm phương pháp Sobel và phương pháp Canny.

*Phân đoạn (Segmentation):* Phân đoạn là một phép xử lý ảnh được sử dụng để phân tách hình ảnh thành các phần khác nhau để phân tích và hiểu hình ảnh. Các phương pháp phân đoạn phổ biến bao gồm phương pháp phân đoạn ngưỡng và phân đoạn dựa trên mô hình.

*Trích xuất đặc trưng (Feature extraction):* Trích xuất đặc trưng là một phép xử lý ảnh được sử dụng để trích xuất các đặc trưng quan trọng trong hình ảnh. Các phương pháp trích xuất đặc trưng phổ biến bao gồm phương pháp HOG (Histogram of Oriented Gradients) và phương pháp SIFT (Scale-Invariant Feature Transform) .

*Phân loại (Classification):* Phân loại là một phép xử lý ảnh được sử dụng để phân loại hình ảnh vào các lớp khác nhau dựa trên các đặc trưng được trích xuất. Các phương pháp phân loại phổ biến bao gồm phương pháp SVM (Support Vector Machine) và phương pháp mạng thần kinh nhân tạo.

Các phép xử lý ảnh trên đây chỉ là một số ví dụ phổ biến. Khi xử lý hình ảnh, chúng ta có thể sử dụng nhiều phương pháp và kết hợp chúng để đạt được kết quả tốt nhất cho một bài toán cụ thể.

### 1.1.6. Ứng dụng của xử lý ảnh

**Bảng 1 Tác dụng của xử lý ảnh trong các lĩnh vực áp dụng**

Các lĩnh vực ứng dụng	Tác dụng
<ul style="list-style-type: none"> <li>- Thông tin ảnh, truyền thông ảnh.</li> <li>- Xử lý ảnh vệ tinh, viễn thám.</li> <li>- Thiên văn, nghiên cứu không gian vũ trụ.</li> </ul>	<ul style="list-style-type: none"> <li>- Cải thiện thông tin hình ảnh cho sự nhận biết của con người.</li> <li>- Cải thiện và phục hồi ảnh.</li> <li>- Dữ liệu vào và ra đều là ảnh.</li> </ul>
<ul style="list-style-type: none"> <li>- Địa chất thăm dò.</li> <li>- Người máy, tự động hóa.</li> <li>- Sinh học, y học.</li> <li>- Vật lý, hóa học.</li> <li>- Giám sát kiểm soát quân sự.</li> <li>- Xử lý ảnh phục vụ cuộc sống</li> </ul>	<ul style="list-style-type: none"> <li>- Trích thông tin từ ảnh cho sự phân tích sâu hơn.</li> <li>- Hiểu ảnh và nhận dạng ảnh.</li> <li>- Đầu vào là ảnh, đầu ra không phải là ảnh mà là biểu diễn nội dung của ảnh như mô tả, giải thích, phân loại.</li> </ul>

## 1.2. Tổng quan về thị giác máy tính

### 1.2.1. Khái niệm

Thị giác máy tính (Computer Vision) là lĩnh vực nghiên cứu và ứng dụng của trí tuệ nhân tạo. Lĩnh vực này liên quan đến việc xử lý và phân tích hình ảnh và video để có thể hiểu và tạo ra thông tin dựa trên những gì được nhìn thấy. Việc phát triển lĩnh vực này có bối cảnh từ việc sao chép các khả năng thị giác của con người bởi sự nhận diện và hiểu biết một hình ảnh mang tính điện tử. Sự nhận diện hình ảnh có thể xem là việc giải quyết vấn đề của các biểu tượng thông tin từ dữ liệu hình ảnh qua cách dùng các mô hình được xây dựng với sự giúp đỡ của các ngành lý thuyết học, thống kê, vật lý và hình học. Thị giác máy tính cũng được mô tả là sự tổng thể của một dải rộng các quá trình tự động và tích hợp và thể hiện cho các nhận thức thị giác.

Thị giác máy tính là một môn học khoa học liên quan đến lý thuyết đằng sau các hệ thống nhân tạo có trích xuất các thông tin từ các hình ảnh. Dữ liệu hình ảnh có thể nhiều dạng, chẳng hạn như dạng chuỗi video, các cảnh từ camera, hay dữ liệu đa chiều từ máy quét y học. Đây còn là một môn học kỹ thuật, trong đó tìm kiếm và áp dụng các mô hình và các lý thuyết cho việc xây dựng các hệ thống thị giác máy tính.

Các lĩnh vực con của thị giác máy tính bao gồm tái cấu trúc cảnh, dò tìm sự kiện, theo dõi video, nhận diện bố cục đối tượng, đánh giá chuyển động, phục hồi ảnh...

Các kỹ thuật phổ biến trong thị giác máy tính bao gồm xử lý ảnh, trích xuất đặc trưng, phân loại và nhận dạng. Các phương pháp khác nhau được sử dụng để giải quyết các vấn đề khác nhau, bao gồm các kỹ thuật dựa trên mô hình, học máy, mạng nơ-ron nhân tạo, và nhiều kỹ thuật khác.

### **1.2.2. Ứng dụng trong thực tiễn**

Ứng dụng của thị giác máy tính rất phong phú và đa dạng, bao gồm nhận dạng khuôn mặt, phát hiện đối tượng, phân tích hình ảnh y sinh, phân tích ảnh vệ tinh, tạo ra thế giới ảo, và nhiều ứng dụng khác. Các ứng dụng này được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm y tế, an ninh, giao thông, sản xuất, giải trí và nhiều lĩnh vực khác.

Một số ứng dụng tiêu biểu của thị giác máy tính bao gồm:

- Nhận dạng khuôn mặt: Sử dụng để xác định một người trong hình ảnh hoặc video.
- Phát hiện đối tượng: Sử dụng để phát hiện các đối tượng trong ảnh hoặc video, bao gồm phát hiện xe cộ, người, động vật, vật thể và nhiều thứ khác.
- Xử lý ảnh y sinh: Sử dụng để phân tích hình ảnh y tế, bao gồm phát hiện ung thư, xác định bệnh và giúp phẫu thuật viên trong quá trình phẫu thuật.

- Phân tích ảnh vệ tinh: Sử dụng để phân tích hình ảnh vệ tinh để giám sát môi trường, dự báo thời tiết, dự đoán đường đi của bão, phát hiện cháy rừng và đánh giá tài nguyên đất đai.
- Tạo ra thế giới ảo: Sử dụng để tạo ra thế giới ảo trong trò chơi điện tử hoặc phim ảnh.
- Nhận diện chữ số viết tay: Ứng dụng trong việc chuyển đổi hình ảnh chữ, số sang văn bản...

Tổng quan về thị giác máy tính là rất đa dạng và phức tạp. Khi sử dụng các kỹ thuật và công cụ thích hợp, thị giác máy tính có thể giúp chúng ta giải quyết nhiều vấn đề phức tạp trong nhiều lĩnh vực khác nhau.



Hình 1.4 Thị giác máy tính

### 1.2.3. Các lĩnh vực liên quan

Các lĩnh vực liên quan của trí tuệ nhân tạo giải quyết các vấn đề như lên kế hoạch tự động hay các suy tính cho các hệ thống robot để dò đường ở một môi trường nào đó. Sự hiểu biết chi tiết của các môi trường này được yêu cầu để dò đường thông qua chúng. Thông tin về môi trường có thể được cung cấp bởi một hệ thống thị giác máy tính, hoạt động như các cảm biến và cung cấp thông tin mức độ cao về môi trường và robot.

Trí tuệ nhân tạo và thị giác máy tính chia sẻ các chủ đề như nhận dạng mẫu và các kỹ thuật học. Kết quả là thị giác máy tính đôi khi được xem là một phần của lĩnh vực trí tuệ nhân tạo hay lĩnh vực khoa học máy tính nói chung.

Thị giác máy tính theo một cách nào đó là sự đảo ngược của đồ họa máy tính. Trong khi đồ họa máy tính sản sinh hình ảnh từ mô hình 3D, thì thị giác máy tính lại thường sản sinh ra các mô hình 3D từ dữ liệu hình ảnh. Có một khuynh hướng kết hợp 2 môn học này, ví dụ như khám phá trong tăng cường thực tế.

### **1.3. Tổng quan về công nghệ nhận dạng biển số xe**

Nhận dạng biển số xe, hay còn gọi là nhận dạng biển số xe tự động (ALPR), là công nghệ cho phép máy tính phát hiện biển số xe và trích xuất thông tin có trong biển số xe đó. Công nghệ này sử dụng AI và các mô hình học máy để xác định được biển số và các kí tự có trong đó.

#### **1.3.1. Lịch sử phát triển**

Do công nghệ nhận dạng biển số xe được ứng dụng phổ biến trong những năm gần đây như nghiên cứu giao thông, quản lý xe cộ tại các bãi đỗ,... nên nhiều người sẽ nghĩ rằng đây là một phát minh gần đây. Nhưng điều bất ngờ là lịch sử của ALPR kéo dài đến thế kỉ trước khi nó được phát minh vào năm 1976 tại chi nhánh phát triển Khoa học Cảnh sát (PSDB) (nay có tên gọi là Chi nhánh phát triển Khoa học bộ Nội vụ) tại Vương quốc Anh và hệ thống đầu tiên được đưa vào sử dụng vào năm 1979.

Các hệ thống thử nghiệm ban đầu đã được triển khai trên Đường A1 tại giao lộ Đường hầm Dartford trên đường cao tốc M25 tại Anh. Vụ bắt giữ đầu tiên được cho là do ANPR phát hiện ra một chiếc ô tô bị đánh cắp vào năm 1981.

Sau đây là những cột mốc phát triển của công nghệ nhận dạng biển số xe:

#### **1993**

ANPR(Automatic Number Plate Recognition) được triển khai lần đầu tiên như một phần của mạng camera “Ring of Steel” xung quanh thành phố Luân

Đôn. Dự án nay là dự án về nhận diện biển số xe lớn nhất tại thời điểm đó và được Cảnh sát Luân Đôn thực hiện nhằm chấm dứt chuỗi vụ đánh bom khủng bố ở khu tài chính IRA.

### **1997**

Trung tâm dữ liệu ANPR Quốc gia của Cảnh sát (NADC) được hình thành như một phần mở rộng của dịch vụ Máy tính Quốc gia của Cảnh sát. Trước khi NADC ra đời, dữ liệu thu thập từ các hệ thống ANPR được giữ lại tại từng lực lượng cảnh sát điều hành các camera, tạo sự phân tán. Tuy nhiên, Trung tâm Dữ liệu ANPR Quốc gia đã cho phép phân tích dữ liệu qua các ranh giới của lực lượng cảnh sát, bằng các tập trung toàn bộ dữ liệu ANPR của Cảnh sát từ khắp nước Anh.

### **2003**

Chương trình The London Congestion Charge được triển khai nhằm mục đích giảm lượng xe cộ trong trung tâm Luân Đôn. Vùng thu phí này, bao gồm 20 km<sup>2</sup>, được vận hành bởi Cơ quan Vận Tải Luân Đôn và bao gồm gần 700 camera ANPR được đặt trên các con đường vào và ra khỏi Vùng thu phí. Đây là lần đầu tiên sử dụng ANPR quy mô lớn như vậy cho mục đích dân sự.

### **2005**

ANPR International Limited được thành lập, tập trung vào phát triển sản phẩm và các giải pháp sử dụng công nghệ ANPR

### **Hiện nay**

ANPR đã trở thành công nghệ phổ biến trên toàn thế giới, được sử dụng trong nhiều lĩnh vực như an ninh giao thông, quản lý bãi đỗ xe, giám sát vi phạm giao thông và nhiều ứng dụng khác.

#### **1.3.2. Khó khăn và thách thức**

Mặc dù được phát minh từ năm 1976 đến nay, bài toán nhận diện biển số xe vẫn gặp những khó khăn và thách thức sau:

- Đa dạng biển số: Mỗi quốc gia hoặc vùng lãnh thổ có hệ thống biển số xe riêng với định dạng và cấu trúc khác nhau. Điều này tạo thành thức lớn khi phải xử lý và nhận dạng nhiều loại biển số xe với nhau.
- Điều kiện ánh sáng và môi trường: Điều kiện ánh sáng không đồng đều có thể khiến chất lượng hình ảnh và khả năng nhận dạng đi xuống
- Góc chụp và kích thước biển số: Khi hình ảnh được chụp từ các góc đa dạng hoặc biển số xe có kích thước nhỏ sẽ có những tác động lớn đến việc nhận dạng chính xác và chuẩn xác.
- Mất đối tượng: Trong một số trường hợp, biển số xe có thể bị che khuất bởi các vật thể khác như cành cây, xe tải hoặc biển quảng cáo. Điều này làm giảm khả năng nhận dạng và đòi hỏi các thuật toán phải xử lý các trường hợp phức tạp.
- Độ phức tạp của hệ thống: Xử lý và phân tích dữ liệu từ hàng trăm hoặc hàng nghìn camera ANPR đòi hỏi một hệ thống phức tạp và mạnh mẽ. Việc xử lý dữ liệu lớn và thời gian thực là một thách thức đối với các hệ thống nhận dạng biển số xe.
- Bảo mật và quyền riêng tư: Vấn đề bảo mật dữ liệu và quyền riêng tư là một thách thức quan trọng trong việc triển khai hệ thống. Cần phải có các biện pháp bảo mật mạnh mẽ để đảm bảo an toàn và sự riêng tư của dữ liệu nhận dạng.

## **1.4. Tổng quan về đề tài**

### **1.4.1. Tên đề tài**

Đề tài: Xây dựng mô hình nhận diện biển số xe sử dụng WPOD.

### **1.4.2. Lý do chọn đề tài**

Nhận diện biển số xe là một bài toán được ứng dụng rất rộng rãi trong xã hội. Nó có thể được ứng dụng trong việc quản lý các phương tiện trong các bãi đỗ xe, hay cũng có thể ứng dụng nó trong việc hỗ trợ những công việc liên quan

đến giao thông vận tải ví dụ như: trạm thu phí không dừng, trích xuất biên số xe của người vi phạm luật giao thông,...

Nhận thấy trong quá trình nhận biên số xe phân kết quả đầu ra đôi khi không được chính xác. Khi tìm hiểu sâu vào, tôi nhận thấy rằng khi sử dụng các thuật toán phát hiện vật thể như YOLO thì kết quả về là một bounding box có hai điểm giới hạn là phía trên bên trái và phía dưới bên phải. Vậy nên là, nếu những biên số xe bị nghiêng, lệch khỏi vị trí chính diện thì bounding box trả về sẽ bị dính thêm những phần ngoại cảnh như đường, chi tiết của xe, ... điều này làm giảm độ chính xác của công việc nhận diện biên số xe đằng sau của hệ thống.



Hình 1.5 Kết quả trả về khi sử dụng YOLO để phát hiện biên số

Vì vậy tôi quyết định thực hiện đề tài **“Xây dựng mô hình nhận diện biên số xe sử dụng WPOD”** để giải quyết được vấn đề biên số xe bị lệch nhằm cải thiện độ chính xác của bài toán nhận diện biên số xe.

#### 1.4.3. Mục tiêu của đề tài

Đề tài **“Xây dựng mô hình nhận diện biên số xe sử dụng WPOD”** có mục tiêu cơ bản là:

- Nghiên cứu mô hình WPOD trong việc đưa biên số xe về hướng chính diện.
- Nắm được các kiến thức cơ bản về Python, về model phát hiện biên số xe và model nhận các kí tự trong biên số xe để train model cũng như cách triển khai nó.



- Ứng dụng WPOD vào hệ thống nhận diện biển số xe để cải thiện độ chính xác của hệ thống.

#### **1.4.4. Đối tượng và phạm vi**

Đối tượng nghiên cứu: Mô hình WPOD và các mô hình phục vụ cho việc phát hiện và nhận diện biển số xe.

Phạm vi nghiên cứu: Dựa vào các bộ dữ liệu được huấn luyện trong các phần của hệ thống ứng dụng nó vào việc nhận diện biển số xe tại điểm khảo sát.

#### **1.4.5. Kết quả dự kiến đạt được**

Nắm vững kiến thức về các thức hoạt động của mô hình WPOD, mô hình phát hiện biển số xe, mô hình nhận diện các kí tự trong biển số xe.

Nắm vững về nguyên lý hoạt động của hệ thống nhận diện biển số xe.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Bài toán nhận diện vật thể

#### 2.1.1. Các phương pháp

Các phương pháp tiên tiến nhất có thể được phân thành hai loại chính: Phương pháp một giai đoạn và phương pháp hai giai đoạn. Các phương pháp một giai đoạn ưu tiên tốc độ suy luận và các mô hình ví dụ bao gồm YOLO, SSD và RetinaNet. Các phương pháp hai giai đoạn ưu tiên độ chính xác phát hiện và các mô hình ví dụ bao gồm Faster R-CNN, Mask R-CNN và Cascade R-CNN.

Hai giai đoạn (Two – Stage): Trong mạng nhiều giai đoạn, nó bắt đầu thực hiện bài toán với việc tìm bounding box dựa trên việc trích xuất khu vực được đề xuất sau đó sử dụng hàm kích hoạt (Activation Function) để phân loại đối tượng và cho kết quả cuối cùng.

Một giai đoạn (One – Stage): Trong các mạng một giai đoạn, chẳng hạn như YOLO, CNN tạo ra các mạng dự đoán cho các vùng trên toàn bộ hình ảnh bằng cách sử dụng các hộp neo (anchor boxer) và các dự đoán được giải mã để tạo các hộp giới hạn (bounding box) cuối cùng cho các đối tượng.

Mỗi phương pháp có những ưu điểm và hạn chế riêng.

- YOLO (You Only Look Once) là một phương pháp phát hiện vật thể nhanh hơn so với phương pháp hai giai đoạn, vì chỉ cần một lần chạy qua ảnh để phát hiện các vật thể. Tuy nhiên, YOLO có độ chính xác thấp hơn so với phương pháp hai giai đoạn.
- SSD (Single Shot MultiBox Detector) là một phương pháp phát hiện vật thể nhanh và độ chính xác cao hơn so với YOLO, nhưng tốn nhiều thời gian tính toán hơn.
- RetinaNet là một phương pháp phát hiện vật thể mới, được giới thiệu bởi tác giả Tsung-Yi Lin vào năm 2017. RetinaNet sử dụng một mạng neural để tạo ra các dự đoán về vị trí và lớp của các vật thể trong một ảnh. Điểm khác biệt của RetinaNet so với các phương

pháp khác là nó sử dụng một cơ chế cân bằng mẫu (Focal Loss) để giảm thiểu tác động của các vật thể chiếm ưu thế (có nhiều mẫu) đối với việc huấn luyện mô hình. RetinaNet đạt được độ chính xác cao và tốc độ xử lý nhanh, phù hợp với các ứng dụng thời gian thực. Tuy nhiên, RetinaNet cần nhiều dữ liệu huấn luyện để đạt được kết quả tốt.

Các phương pháp hai giai đoạn như Fast R-CNN, Mask R-CNN, Cascade R-CNN là các phương pháp ưu tiên độ chính xác. Những phương pháp này tìm kiếm các vùng đáng chú ý trong ảnh trước khi phân loại và phân đoạn các vật thể trong các vùng đó. Phương pháp này đạt độ chính xác nhưng tốn nhiều thời gian tính toán và không phù hợp với các ứng dụng thời gian thực.

- Fast R-CNN là một phương pháp nhận diện vật thể nhanh và chính xác sử dụng mạng neural, được phát triển bởi Ross Girshick vào năm 2015. Nó là một phiên bản nâng cấp của R-CNN và Faster R-CNN. Ứng dụng của Fast R-CNN là phân loại và nhận diện vật thể trong ảnh và video.
- Mask R-CNN là một phương pháp nhận diện vật thể và phân đoạn ảnh sử dụng mạng neural, được phát triển bởi tác giả Kaiming He vào năm 2017. Nó là một phiên bản nâng cấp của Faster R-CNN và cho phép nhận diện vật thể và phân đoạn ảnh cùng một lúc.
- Cascade R-CNN là một phương pháp nhận diện vật thể sử dụng mạng neural, được phát triển bởi tác giả Zhaowei Cai vào năm 2018. Nó là một phiên bản nâng cấp của Faster R-CNN và sử dụng một loạt các mô hình nhận diện vật thể để tăng độ chính xác của hệ thống.

### **2.1.2. Phương pháp đánh giá mô hình**

Độ chính xác trung bình trung bình (mAP) là chỉ số được sử dụng để đánh giá các mô hình phát hiện đối tượng như Fast R-CNN, YOLO, Mask R-CNN,...

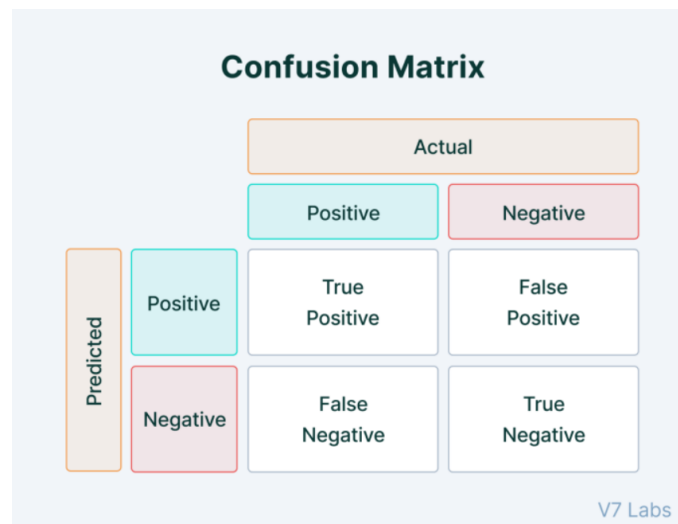
Giá trị trung bình của độ chính xác trung bình (AP) được tính dựa trên các giá trị recall từ 0 đến 1.

Công thức mAP dựa trên các chỉ số phụ sau:

- Ma trận hỗn loạn (Confusion Matrix)
- Giao trên hợp (IoU)
- Recall
- Độ chính xác (Precision)

Có bốn thuộc tính để tạo ma trận hỗn loạn:

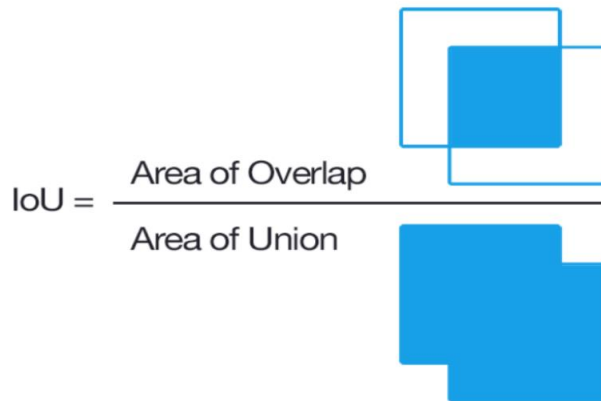
- Dương tính thật (TP): Mô hình đã dự đoán nhãn và khớp với nhãn thực tế.
- Âm tính thật (TN): Mô hình không dự đoán nhãn và không phải là một phần của nhãn thực tế.
- Dương tính giả (FP): Mô hình đã dự đoán một nhãn, nhưng nó không phải là một phần của nhãn thực tế (Lỗi loại I) .
- Âm tính giả (FN): Mô hình không dự đoán nhãn, nhưng nó là một phần của nhãn thực tế (Lỗi loại II) .



Hình 2.1 Ma trận hỗn loạn

Giao trên hợp (IOU): là tỷ lệ giữa đo lường mức độ giao nhau giữa hai đường bao (thường là đường bao dự đoán và đường bao thực) . Tỷ lệ này được

tính dựa theo phần diện tích giao nhau giữa 2 đường bao với phần tổng diện tích giao nhau và không giao nhau giữa chúng.



Hình 2.2 Intersection of union

Tiêu chí theo ngưỡng tự đặt:

- $IOU \geq \text{ngưỡng}$ : TP – True Positive
- $IOU < \text{ngưỡng}$ : FP – False Positive
- Đối tượng không được nhận dạng: FN- False Negative

Độ chính xác(Precision): tỉ lệ số lượng dự đoán đúng các trường hợp True Positive trên tổng số dự đoán Positive.

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

Recall: tỉ lệ số lượng dự đoán đúng các trường hợp True Positive trên tổng số các trường hợp thực tế là Positive.

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

Cách tính mAP: tìm Độ chính xác trung bình (AP) cho mỗi lớp và sau đó lấy trung bình trên mỗi lớp.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

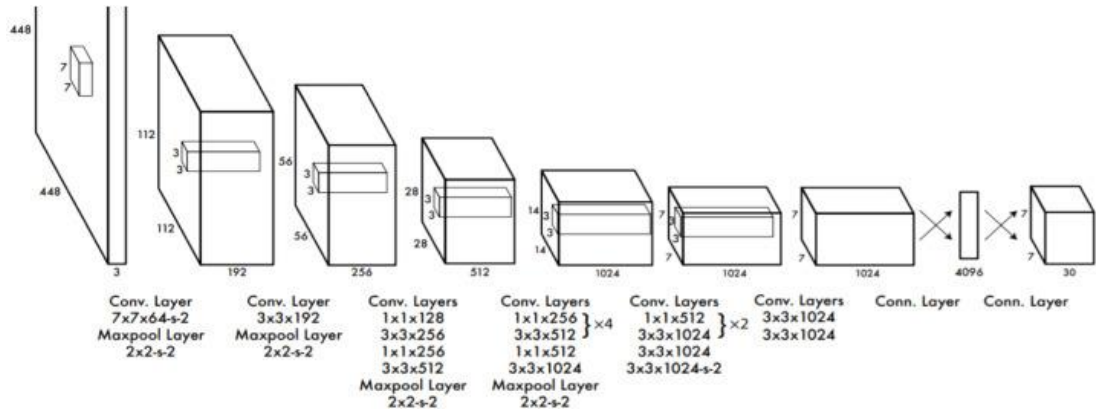
Chỉ số mAP kết hợp với sự đánh đổi giữa độ chính xác và recall, đồng thời xem xét cả kết quả dương tính giả(FP) và âm tính giả (FN) . Thuộc tính này làm cho mAP trở thành thước đo phù hợp cho hầu hết các ứng dụng phát hiện.

## 2.2. Lịch sử phát triển YOLO

YOLO là thuật toán phát hiện vật thể thời gian thực một giai đoạn tối tân nhất (SOTA) . Bất kỳ lập trình viên đam mê thị giác máy tính chắc chắn đã nghe nói về các mô hình YOLO để phát hiện vật thể. Kể từ khi YOLO đầu tiên được giới thiệu vào năm 2015, nó đã thu hút được rất nhiều sự yêu thích trong cộng đồng thị giác máy tính. Sau đó, nhiều phiên bản YOLO và mới nhất là YOLOv8 đã được phát hành bởi những người khác nhau. Các cải tiến theo từng phiên bản của YOLO được đánh giá rất cao và quan trọng trong lĩnh vực Deep Learning. Trên phương diện phát triển của công nghệ này, các phiên bản được nâng cấp bằng các kỹ thuật mới để giúp chính YOLO ngày càng hoàn thiện trong tương lai.

- Mô hình YOLO đầu tiên được giới thiệu bởi Joseph Redmon và tất cả trong bài báo năm 2015 với tiêu đề “*You Only Look Once: Unified, Real-Time Object Detection*” [1]. YOLO được tạo ra với mục tiêu loại bỏ thuật toán nhiều giai đoạn và thực hiện phát hiện đối tượng chỉ trong một giai đoạn duy nhất, do đó cải thiện thời gian suy luận. YOLOv1 là phiên bản đầu tiên của YOLO. Nó là một trong những phương pháp đầu tiên sử dụng mạng neural để nhận diện vật thể. YOLOv1 có tốc độ xử lý nhanh và độ chính xác tương đối cao, nhưng vẫn còn nhiều hạn chế về khả năng phát hiện các vật thể nhỏ hoặc gần nhau.
  - YOLOv1 chia hình ảnh đầu vào thành một lưới ô vuông và mỗi ô sẽ dự đoán các đối tượng có khả năng xuất hiện trong ô đó.
  - Sử dụng phân lớp đa nhãn (multi-label classification): YOLOv1 có thể dự đoán nhiều đối tượng khác nhau trong cùng một ô.
  - Sử dụng hàm mất mát tổng quát (generalized IoU loss): YOLOv1 sử dụng hàm mất mát tổng quát để đánh giá độ chính xác của mô hình.

- Sử dụng kết hợp giữa phân lớp và dự đoán vị trí (objectness score): YOLOv1 sử dụng một điểm số (objectness score) để đánh giá khả năng xuất hiện của đối tượng trong ô.



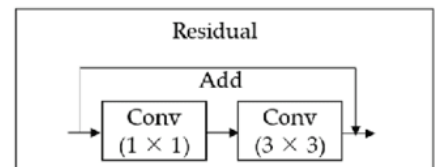
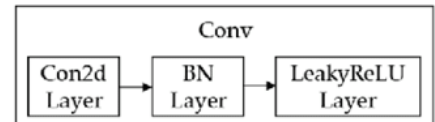
Hình 2.3 Kiến trúc mạng YOLOv1

- YOLOv2 được phát hành bởi Joseph Redmon và Ali Farhadi vào năm 2016 trong bài báo của họ có tiêu đề “YOLO9000: Better, Faster, Stronger” [2]. 9000 biểu thị rằng YOLOv2 có thể phát hiện hơn 9000 loại đối tượng. YOLOv2 có tốc độ xử lý nhanh hơn và độ chính xác cao hơn so với YOLOv1. Nó cũng có khả năng phát hiện các vật thể nhỏ hơn và gần nhau hơn.
  - YOLOv2 sử dụng kỹ thuật batch normalization để giảm độ lệch và tăng độ chính xác của mô hình.
  - YOLOv2 sử dụng một mạng nơ-ron tích chập (convolutional neural network) có cấu trúc mới để tăng tốc độ xử lý.
  - YOLOv2 sử dụng kỹ thuật anchor boxes để tăng độ chính xác của việc dự đoán vị trí đối tượng.
  - YOLOv2 sử dụng kỹ thuật softmax để tăng độ chính xác của việc phân loại đối tượng.
- Vào năm 2018, Joseph Redmon và Ali Farhadi đã giới thiệu phiên bản thứ ba của YOLOv3 trong bài viết “YOLOv3: An Incremental Improvement” [3]. Một trình trích xuất đặc trưng tốt hơn đã được giới thiệu cùng với việc 53 lớp tích chập được đào tạo trên

ImageNet. Độ chính xác của YOLOv3 tốt hơn YOLOv2 nhưng chậm hơn YOLO v2 do có nhiều lớp hơn.

- Backbone network được cải tiến để tăng độ chính xác và tốc độ xử lý bằng mô hình Darknet-53.
- Sử dụng kỹ thuật skip connection để kết hợp thông tin từ các tầng khác nhau của mạng neural, giúp cải thiện khả năng phát hiện các vật thể nhỏ và gần nhau.
- Sử dụng kỹ thuật kết hợp đa tầng để tăng độ chính xác của việc phân loại lớp vật thể.
- YOLOv3 còn hỗ trợ phát hiện các vật thể trong các tình huống khó như các vật thể bị che khuất hoặc có độ lệch góc lớn.

Layer	Filters size	Repeat	Output size
Image			$416 \times 416$
Conv	$32 \ 3 \times 3/1$	1	$416 \times 416$
Conv	$64 \ 3 \times 3/2$	1	$208 \times 208$
Conv	$32 \ 1 \times 1/1$	<div style="border: 1px dashed black; padding: 2px; display: inline-block;"> Conv Conv Residual </div> $\times 1$	$208 \times 208$
Conv	$64 \ 3 \times 3/1$		$208 \times 208$
Residual			$208 \times 208$
Conv	$128 \ 3 \times 3/2$	1	$104 \times 104$
Conv	$64 \ 1 \times 1/1$	<div style="border: 1px dashed black; padding: 2px; display: inline-block;"> Conv Conv Residual </div> $\times 2$	$104 \times 104$
Conv	$128 \ 3 \times 3/1$		$104 \times 104$
Residual			$104 \times 104$
Conv	$256 \ 3 \times 3/2$	1	$52 \times 52$
Conv	$128 \ 1 \times 1/1$	<div style="border: 1px dashed black; padding: 2px; display: inline-block;"> Conv Conv Residual </div> $\times 8$	$52 \times 52$
Conv	$256 \ 3 \times 3/1$		$52 \times 52$
Residual			$52 \times 52$
Conv	$512 \ 3 \times 3/2$	1	$26 \times 26$
Conv	$256 \ 1 \times 1/1$	<div style="border: 1px dashed black; padding: 2px; display: inline-block;"> Conv Conv Residual </div> $\times 8$	$26 \times 26$
Conv	$512 \ 3 \times 3/1$		$26 \times 26$
Residual			$26 \times 26$
Conv	$1024 \ 3 \times 3/2$	1	$13 \times 13$
Conv	$512 \ 1 \times 1/1$	<div style="border: 1px dashed black; padding: 2px; display: inline-block;"> Conv Conv Residual </div> $\times 4$	$13 \times 13$
Conv	$1024 \ 3 \times 3/1$		$13 \times 13$
Residual			$13 \times 13$



Hình 2.4 Kiến trúc mạng YOLOv3

- YOLOv4 không phải do Joseph Redmon phát hành mà bởi Alexey Bochkovskiy, v. v. trong bài báo năm 2020 của họ “YOLOv4: Optimal Speed and Accuracy of Object Detection” [4]. YOLOv4 so với các phiên bản trước đó, có nhiều cải tiến đáng kể như:



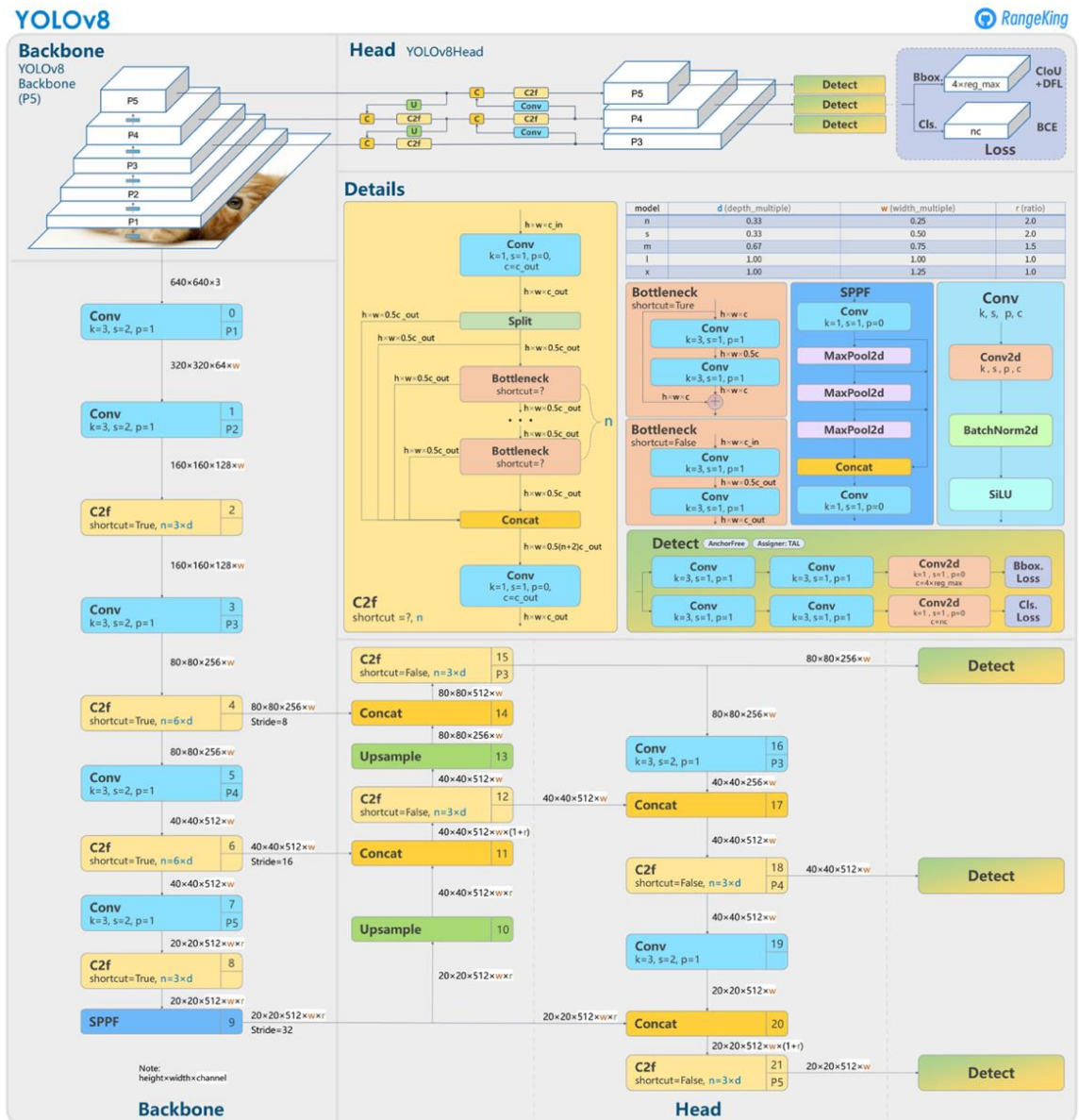
- Sử dụng mô hình CSPDarknet sử dụng kỹ thuật kết hợp các lớp convolution giảm số lượng tham số và tăng khả năng học tập, mô hình này giúp YOLOv4 giảm hiện tượng overfitting, tăng tốc độ xử lý và độ chính xác cao hơn.
- Sử dụng kỹ thuật SPP (Spatial Pyramid Pooling) để giảm thiểu số lượng tính toán và tăng độ chính xác.
- Sử dụng kỹ thuật PAN (Path Aggregation Network) để kết hợp thông tin từ các tầng khác nhau của mạng neural, giúp cải thiện khả năng phát hiện các vật thể nhỏ và gần nhau.
- Sử dụng kỹ thuật YOLOv4 Neck để tăng độ chính xác của việc phân loại lớp vật thể.
- Hỗ trợ phát hiện các vật thể trong các tình huống khó như các vật thể bị che khuất hoặc có độ lệch góc lớn.
- YOLOv5 [5] được cho là thành viên tiếp theo của gia đình YOLO được phát hành vào năm 2020 bởi công ty Ultralytics chỉ vài ngày sau YOLOv4. YOLOv5 có nhiều điểm khác biệt so với phiên bản trước đó là YOLOv4:
  - Tốc độ xử lý và độ chính xác cao hơn.
  - Sử dụng mô hình EfficientNet để tăng tốc độ xử lý và độ chính xác, trong khi YOLOv4 sử dụng mô hình CSPDarknet.
  - YOLOv5 sử dụng kỹ thuật BiFPN (Bidirectional Feature Pyramid Network) để kết hợp thông tin từ các tầng khác nhau của mạng neural, giúp cải thiện khả năng phát hiện các vật thể nhỏ và gần nhau, trong khi YOLOv4 sử dụng kỹ thuật PAN.
  - YOLOv5 có khả năng phát hiện các vật thể nhỏ và gần nhau tốt hơn so với YOLOv4.
  - YOLOv5 có tính năng tự động tinh chỉnh các siêu tham số để tối ưu hóa độ chính xác và tốc độ xử lý.

- Trong năm 2022, Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei đã phát hành YOLOv6 trong bài báo “*YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*” [6]. YOLOv6 là phiên bản cũng được đánh giá có bước tiến lớn, so với YOLOv5 thì nó vượt trội hơn hẳn ở các mô hình và kỹ thuật cải tiến dựa trên ý tưởng thiết kế mạng thần kinh nhận biết phân cứng:
  - Mô hình không máy dò neo có khả năng khái quát hóa mạnh mẽ và logic mã đơn giản cải thiện hơn về tốc độ.
  - Chính sách gán thẻ SimOTA tự động phân bổ cải thiện độ chính xác.
  - Hộp hồi quy giới hạn mát mát SIOU giám sát quá trình học của mạng cải thiện độ chính xác hồi quy.
  - YOLOv6 có thể phát hiện các vật thể nhỏ hơn một cách đáng tin cậy hơn, nhưng nó lại nhạy cảm với nhiễu so với YOLOv5 và gặp khó khăn với các vật thể ở gần. So với YOLOv5, MT-YOLOv6 thiếu tính ổn định nhưng bù lại khả năng phát hiện đối tượng nhỏ trong môi trường đông đúc rất ấn tượng. Về tính linh hoạt, YOLOv5 sử dụng YAML và YOLOv6 xác định các tham số mô hình trực tiếp bằng Python. Người ta nhận thấy rằng YOLOv5 có thể tùy chỉnh nhiều hơn YOLOv6.
- Cùng năm, YOLOv7 được phát hành bởi Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao với bài báo “*YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*” [7]. YOLOv7 là mô hình cơ bản được tối ưu hóa cho điện toán GPU thông thường, cũng có cải thiện về tốc độ và độ

chính xác bằng việc cải cách một số kiến trúc. Tương tự như YOLOv4 được chia tỷ lệ, các xương sống YOLOv7 không sử dụng các xương sống được đào tạo trước ImageNet. Thay vào đó, các mô hình được đào tạo hoàn toàn bằng bộ dữ liệu COCO.

- Cải cách mạng E-ELAN (Mạng tổng hợp lớp hiệu quả mở rộng) , nó là khối tính toán lấy cảm hứng từ nghiên cứu trước đây về hiệu quả mạng và đã được thiết kế bằng cách phân tích các yếu tố ảnh hưởng đến tốc độ, độ chính xác về chi phí truy cập bộ nhớ, tỷ lệ đầu vào và đầu ra, yếu tố hoạt động khôn ngoan, khả năng kích hoạt, đường đi đạo hàm.
- Mở rộng các mô hình dựa trên phép tìm kiếm kiến trúc mạng NAS (Network Architecture Search) có yếu tố tỉ lệ và độc lập.
- Trainable BoF (Bag of Freebies) là một kỹ thuật giám sát sâu có khuôn khổ đa đầu, có thể sử dụng huấn luyện để tích hợp tái tham số hóa theo kế hoạch và huấn luyện thô cho việc phụ trợ trong mô hình học tập và phạt nếu làm mất đầu chỉ- đầu đầu tiên trong dãy các đầu dự đoán. Bỏ trợ công cụ gán nhãn mềm thay vì nhãn cứng phù hợp cho mức độ học tập mạnh của mô hình.
- Vào ngày 10 tháng 1 năm 2023, Ultralytics tiếp tục cho ra mắt phiên bản mới nhất của YOLO là YOLOv8 [8] khẳng định những tiến bộ về cấu trúc và thay đổi kiến trúc với kết quả tốt hơn. Những điểm mới có trong YOLO v8 :
  - Phát hiện không hộp neo: mô hình dự đoán trực tiếp tâm của một đối tượng thay vì phân bù (offset) từ hộp neo đã biết.
  - Các lớp tích chập mới: lớp tích chập 6x6 đầu tiên được thay thế bằng một lớp 3x3, khối xây dựng chính được thay đổi, lớp C2f đã thay thế lớp C3. Mô đun được tổng hợp như hình

bên dưới, trong đó “f” là số đặc trưng, “e” là tỉ lệ mở rộng và CBS là một khối bao gồm một lớp tích chập, một lớp BatchNorm và một lớp SiLU.



Hình 2.5 Kiến trúc mạng YOLOv8

- Vào ngày 29 tháng 2 năm 2024, YOLOv9 được phát hành bởi Chien-Yao Wang, I-Hau Yeh và Hong – Yuan mark Liao với bài báo “*Yolov9: Learning What You want to Learn Using Programmable Gradient Information*”[9]. YOLOv9 có hai điểm nhấn, cũng chính là hai thành phần cấu thành bao gồm:

- Programmable Gradient Information(PGI): Kỹ thuật được đề xuất nhằm khắc phục vấn đề mất mát thông tin của dữ liệu đầu vào khi trải qua quá trình trích xuất đặc trưng và biến đổi không gian từ layer này qua layer khác trong các mạng deep learning. Vấn đề này còn được biết đến với tên gọi là Information Bottleneck. PGI cho thấy độ hiệu quả vượt qua các giải pháp hiện có như Reversible Architecture, Masked Modeling, Deep Supervision.
- GELAN(Generalized Efficient Layer Aggregation Network): Mô hình được nhóm tác giả đề xuất là sự kết hợp giữa hai kiến trúc mạng CSPNet và ELAN, đều được thiết kế dựa trên kỹ thuật Gradient Path Planning. Một điểm điểm quan trọng được nhấn mạnh bởi tác giả là khả năng của GELAN là sự tối ưu về việc sử dụng trọng số đã giúp GELAN nhẹ hơn các mô hình khác, dẫn đến đạt được tốc độ inference vượt trội song vẫn giữ được độ chính xác cao.

Trong đề tài này, tôi đề xuất sử dụng YOLOv9 để phát hiện biển số xe.

### **2.3. Nhận dạng biển số xe**

Bài toán nhận dạng biển số xe là một bài toán điển hình dựa trên các phương pháp xử lý ảnh và cả những kỹ thuật sử dụng Deep Learning, trong đó tiêu biểu là bài toán nhận diện ký tự quang học (Optical Character Recognition). Bài toán nhận dạng biển số xe sẽ tập trung vào việc phát hiện các chữ số trong một biển số xe gắn máy. Bản chất của mô hình này là một bài toán phát hiện các ký tự bao gồm các chữ cái viết hoa, viết thường, chữ số, và các ký tự đặc biệt có trên biển số xe máy. Ban đầu mô hình thu nhận biển số xe, trích xuất từng ký tự trong biển số, sau quá trình tính toán, mô hình sẽ đưa ra dự đoán xem các ký tự đó là gì, ghép chúng lại và đưa ra dự đoán hoàn chỉnh biển số xe đó.

Dựa trên phương pháp nhận dạng ký tự quang học (OCR) là một phương pháp nhận diện các ký tự quang học, có tính phổ biến và kết quả đạt được tương

đối khả quan, đã có nhiều mô hình nhận dạng biển số xe ra đời như *OpenALPR* [10], *CR-NET* [11].

Trong những năm gần đây, nhiều nhà nghiên cứu đã giải quyết vấn đề phát hiện biển số xe cho phương tiện. Phát hiện biển số là một trong những bước quan trọng trong tự động nhận dạng biển số vì việc phát hiện chính xác biển số liên quan đến độ chính xác của phân đoạn và các giai đoạn nhận dạng. Một trong những tính năng phân biệt được sử dụng trong phát hiện biển số là hình dạng hình học của nó với tỷ lệ khung hình đã biết. Trong bài báo “*Automatic number plate recognition system*” [12] và “*An efficient method of license plate location*” [13], toán tử Sobel dọc đã được áp dụng để phát hiện các cạnh dọc, sau đó là xác minh biển số bằng cách sử dụng tỷ lệ khung hình chiều rộng trên chiều cao. Phương pháp tiếp cận dựa trên biên nhạy cảm hơn với các cạnh không mong muốn “*Automatic license plate recognition (ALPR): A state-of-the-art review*” [14]. Một số biển số có màu sắc khác nhau để phân biệt chủ sở hữu của phương tiện. Shi và cộng sự. Bài báo “*Automatic license plate recognition system based on color image processing*” [15] đã đề xuất một mô hình HSI (màu sắc, độ bão hòa, cường độ) để phát hiện biển số vì các mô hình màu này không nhạy cảm với các ánh sáng khác nhau.

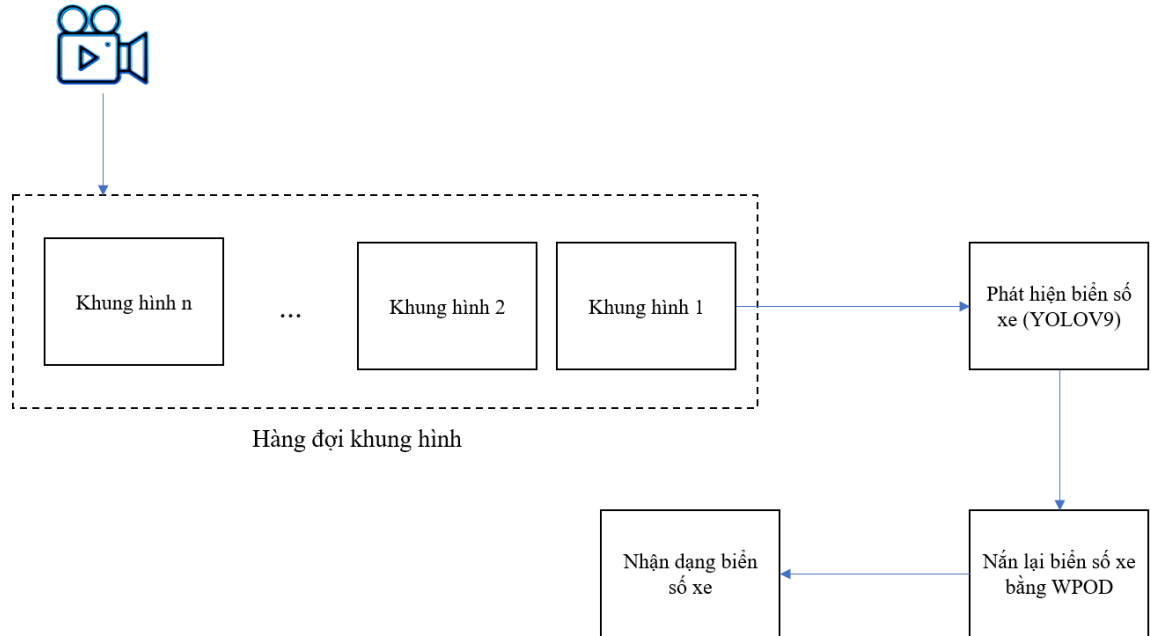
Trong bài nghiên cứu của Bùi Quang Mạnh với tiêu đề “*Nhận diện biển số xe Việt Nam*” [16] đã sử dụng những kỹ thuật bao gồm 4 bước: xác định tọa độ vùng chứa biển số xe sử dụng mô hình YOLOv3 tiny, sử dụng thuật toán Segment để tách từng ký tự trên hình ảnh biển số xe, xây dựng mô hình CNN để phân loại các ký tự, định dạng lại biển số xe bằng cách ghép lại các ký tự cùng hàng với nhau.

Trong đề tài này, tôi đã áp dụng những kỹ thuật được sử dụng trong nghiên cứu của Bùi Quang Mạnh và sử dụng mô hình CNN để nhận dạng ký tự từ hình ảnh biển số xe.

## CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

### 3.1. Tổng quan về hệ thống

Trong đề tài này, nhận diện biển số xe được mô tả như sau:



Hình 3.1 Tổng quan về hệ thống

Mô hình được mô tả như sau:

- Từng frame được ghi lại bằng camera và đi vào hàng đợi khung hình
- Mỗi frame được đi qua mô hình phát hiện biển số xe và trả về tọa độ của các hộp giới hạn (bounding box) chứa biển số xe.
- Hệ thống nhận được được tọa độ của hộp giới hạn sẽ tiến hành mở rộng hình ảnh và nắn lại biển số xe bằng WPOD về phía chính diện.
- Hình ảnh sau khi được nắn lại sẽ được đưa vào nhận dạng biển số xe để trích xuất thông tin về biển số và được hiển thị lại lên ảnh.

### 3.2. Xây dựng hệ thống

Hình ảnh đầu vào là hình ảnh biển số xe sẽ được đưa vào mô hình nhận dạng biển số xe dựa trên thuật toán nhận dạng kí tự quang học (OCR) . Tôi đã áp dụng những kĩ thuật trong bài nghiên cứu của Bùi Quang Mạnh với tiêu đề “*Nhận diện biển số xe Việt Nam*”[16] đã sử dụng những kĩ thuật gồm 4 bước:

xác định tọa độ vùng chứa biển số xe sử dụng mô hình YOLOv3 tiny, sử dụng thuật toán Segment để tách từng ký tự trên hình ảnh biển số xe, xây dựng mô hình CNN để phân loại các ký tự, định dạng lại biển số xe bằng cách ghép lại các ký tự cùng hàng với nhau. Nhờ áp dụng mô hình phát hiện biển số xe trong bài báo “*License Plate Detection and Recognition in Unconstrained Scenarios*” [11], tọa độ của biển số xe vị phạm đã được ghi lại và uốn nắn hình ảnh cùng lúc giúp tăng khả năng nhận dạng ký tự nhưng trong quá trình thực hiện, tôi nhận ra rằng mô hình này nhận dạng rất kém nếu có nhiều vật thể, nên tôi đã quyết định cấu trúc của bài toán nhận diện biển số xe này gồm 5 bước:

- Bước 1: Xác định vùng chứa biển số sử dụng YOLOv9.
- Bước 2: Sử dụng WPOD-NET xác định lại biển số xe trong các bounding box(đã mở rộng ảnh) mà YOLO v9 đã phát hiện được.
- Bước 3: Sử dụng thuật toán Segment để phân đoạn từng ký tự trên biển số.
- Bước 4: Xây dựng một model CNN để phân loại từng ký tự.
- Bước 5: Định dạng lại biển số.

### **3.2.1. Xác định vùng chứa biển số xe sử dụng YOLOv9**

YOLOv9 nổi bật với cách tiếp cận sáng tạo để vượt qua những thách thức mất thông tin vốn có trong các mạng nơ-ron sâu. Bằng cách tích hợp PGI và kiến trúc GLAN linh hoạt. YOLOv9 không chỉ nâng cao năng lực học tập của mô hình mà còn đảm bảo lưu giữ thông tin quan trọng trong suốt quá trình phát hiện vật thể, do đó đạt được độ chính xác và hiệu suất vượt trội.

#### **Những cải tiến cốt lõi của YOLOv9**

Những cải tiến của YOLOv9 bắt nguồn từ việc giải quyết những thách thức do mất thông tin trong các mạng nơ-ron sâu. Nguyên tắc tắc nghẽn thông tin và sử dụng sáng tạo các chức năng đảo ngược là trọng tâm trong thiết kế của nó, đảm bảo YOLOv9 duy trì hiệu quả và độ chính xác cao.

- Nguyên tắc tắc nghẽn thông tin cho thấy một thách thức cơ bản trong học sâu: khi dữ liệu đi qua các lớp liên tiếp của mạng, khả



năng mất thông tin tăng lên. YOLOv9 chống lại thách thức này bằng cách triển khai Programmable Gradient information (PGI), nó giúp hỗ trợ bảo quản dữ liệu thiết yếu trên độ sâu của mạng, đảm bảo gradient đáng tin cậy hơn và do đó, hội tụ mô hình và hiệu suất trở lên tốt hơn.

- Chức năng đảo ngược là một nền tảng khác trong thiết kế của YOLOv9. YOLOv9 kết hợp các chức năng đảo ngược trong kiến trúc của nó để giảm thiểu nguy cơ suy thoái thông tin, đặc biệt là ở các lớp sâu hơn, đảm bảo bảo toàn dữ liệu quan trọng cho các nhiệm vụ phát hiện đối tượng.
- YOLOv9 giải quyết mất thông tin đặc biệt quan trọng đối với các mô hình nhẹ, thường không được tham số hóa và dễ bị mất thông tin quan trọng trong quá trình chuyển tiếp. Nhờ vào việc sử dụng PGI và các chức năng đảo ngược, nó đảm bảo rằng ngay cả với một mô hình được sắp xếp hợp lý, thông tin cần thiết để phát hiện đối tượng chính xác vẫn được giữ lại và sử dụng hiệu quả.

### Hiệu suất của YOLOv9 trên tập dữ liệu MS COCO:

Bảng 2 Hiệu suất của YOLOv9 trên tập dữ liệu MS COCO

Detection (COCO)		Segmentation (COCO)			
Model	size (pixels)	mAP <sup>val</sup> <sub>50-95</sub>	mAP <sup>val</sup> <sub>50</sub>	params (M)	FLOPs (B)
YOLOv9t	640	38.3	53.1	2.0	7.7
YOLOv9s	640	46.8	63.4	7.2	26.7
YOLOv9m	640	51.4	68.1	20.1	76.8
YOLOv9c	640	53.0	70.2	25.5	102.8
YOLOv9e	640	55.6	72.8	58.1	192.5

Các vòng lặp của YOLOv9, từ model YOLOv9t đến YOLOv9e cho thấy được sự cải thiện không chỉ về độ chính xác (mAP) mà còn về độ hiệu quả với số lượng tham số và nhu cầu tính toán (FLOP) giảm. Điều này cho thấy rằng

YOLO cung cấp độ chính xác cao trong khi vẫn duy trì hoặc giảm chi phí tính toán so với các phiên bản tiền nhiệm và các mô hình cạnh tranh.

YOLOv9 thể hiện được những lợi ích đáng chú ý như sau:

- Mô hình nhẹ: YOLOv9s vượt qua YOLO MS-S về hiệu suất tham số và tải tích toán trong khi đạt được sự cải thiện từ 0.4 - 0.6% trong AP
- Các mô hình từ trung bình đến lớn: YOLOv9m và YOLOv9e cho thấy những tiến bộ đáng chú ý trong việc cân bằng sự cân bằng giữa độ phức tạp của mô hình và hiệu suất phát hiện, giúp giảm đáng kể số lượng tham số và số lượng tính toán trong khi độ chính xác được cải thiện.

Mô hình YOLOv9c, đặc biệt nhấn mạnh sự hiệu quả của việc tối ưu hóa kiến trúc. Với số lượng tham số ít hơn 42% và số lượng tính toán yêu cầu ít hơn 21% so với YOLOv7 AF, nhưng nó đạt được độ chính xác tương đương, thể hiện những cải tiến hiệu quả đáng kể của YOLOv9.

Những kết quả này cho thấy những tiến bộ chiến lược của YOLOv9 trong thiết kế mô hình, nhấn mạnh hiệu quả nâng cao của nó mà không ảnh hưởng đến độ chính xác cần thiết cho các nhiệm vụ phát hiện đối tượng trong thời gian thực. Mô hình này không chỉ đẩy mạnh ranh giới giữa các chỉ số hiệu suất mà còn nhấn mạnh tầm quan trọng của hiệu quả tính toán, làm cho nó trở thành một sự phát triển quan trọng trong lĩnh vực thị giác máy tính.

Trong đề tài này, tôi quyết định sử dụng mô hình YOLOv9c để phát hiện biển số xe.

### **Phương pháp thực hiện**

Mô hình YOLOv9 được sử dụng để xác định vị trí các hộp giới hạn (bounding box) của biển số xe của các phương tiện lưu thông trên đường. Mô hình phát hiện được vật thể và sẽ trả về tọa độ của các bounding box trên khung hình gồm có vị trí phía trên bên trái và phía dưới bên phải.

Giả sử mô hình trả về, tọa độ 2 điểm của hộp giới hạn của biển số xe là

- Phía trên bên trái:  $x_0, y_0$
- Phía dưới bên phải:  $x_1, y_1$

Ta sẽ tiến hành mở rộng ảnh để làm đầu vào cho mạng WPOD, có 2 cách để mở rộng ảnh:

- Cách 1: từ các bounding box đã cho ta sẽ cộng vào mỗi tọa độ một giá trị nguyên để mở rộng ảnh.
- Cách 2: ta các bounding box đã cho ta sẽ cộng vào mỗi tọa độ một giá trị tỉ lệ với kích thước của bounding box.

Sau khi cân nhắc kĩ lưỡng, trong đề tài này tôi sử dụng cách 2 để mở rộng ảnh cho đầu vào của WPOD.

Đầu tiên, tôi tính chiều rộng và chiều cao của boundingbox như sau:

$$w_{bb} = x_1 - x_0$$

$$h_{bb} = y_1 - y_0$$

Giá trị mở rộng cho chiều rộng và chiều cao lần lượt được tính như sau:

$$\text{delta}_x = k_x * w_{bb}$$

$$\text{delta}_y = k_y * h_{bb}$$

Trong quá trình thực hiện đề tài, tôi có thử đặt một số hằng số mở rộng sao cho khi sử dụng WPOD-Net sẽ có thể nhận diện ra biển số, tôi đã thử các giá trị trên bộ dữ liệu gồm có 350 ảnh:

**Bảng 3 Kết quả thực nghiệm xác định hằng số mở rộng  $k_x$  và  $k_y$**

$K_x$	$K_y$	Tỷ lệ phát hiện	Nhận xét
0.5	0.5	146/350	Các bounding box chỉ chứa một phần của biển số xe.
1	1	305/350	Không nhận diện được biển số xe một dòng.
1.5	1.5	323/350	Vẫn không nhận diện được biển số xe một dòng.
1.9	1.9	332/350	Nhận diện được 1 phần của biển số xe một dòng.

2	2	330/350	Vẫn chưa nhận diện được hoàn toàn biển số xe một dòng.
2. 6	2. 6	324/350	Bounding box chỉ chứa được khoảng 70-80% biển số xe 1 dòng.
2. 9	2. 9	313/350	Nhận diện được hoàn toàn biển số xe một dòng nhưng biển số xe hai dòng nhận diện bị lỗi.
1. 5	2. 9	322/350	Một số bounding box phát hiện vẫn còn bị sót một vài phần của biển số.
1. 6	2. 9	324/350	Đã nhận diện được cả 2 loại biển.
1. 7	2. 9	326/350	Đã nhận diện được cả 2 loại biển.
1. 9	2. 9	329/350	Đã nhận diện được cả 2 loại biển, có một số ảnh phát hiện biển số bị sai.
2	2. 9	332/350	Đã nhận diện được cả 2 loại biển, có một số ảnh phát hiện biển số bị sai.
2. 1	2. 9	331/350	Một số biển số xe một dòng bắt đầu chỉ nhận diện được 90% biển.

Kết luận sau khi kết thúc thực nghiệm, ta thấy với giá trị  $k_x = 1.7$  và  $k_y = 2.9$  cho ra kết quả tốt nhất mặc dù có một số ảnh vẫn còn thiếu 1 vài chi tiết nhưng ta vẫn có thể chấp nhận được, nên ta sẽ lấy 2 giá trị này là hằng số mở rộng cho đề tài này.

Một số kết quả thực nghiệm:



Hình 3.2 Kết quả thực nghiệm với  $k_x = k_y = 0.5$



Hình 3.3 Kết quả thực nghiệm với  $k_x = k_y = 1.9$



Hình 3.4 Kết quả thực nghiệm với  $k_x = k_y = 2.9$



Hình 3.5 Kết quả thực nghiệm với  $k_x = 1.7$  và  $k_y = 2.9$

Với hằng số mở rộng  $k_x = 1.7$  và  $k_y = 2.9$ , ta phải kiểm tra xem liệu bounding box này có thể mở rộng được không, tại vì nếu mở rộng ra ngoài của ảnh gốc thì ảnh sau khi cắt từ ảnh gốc sẽ bị lỗi.

Ta kiểm tra như sau:

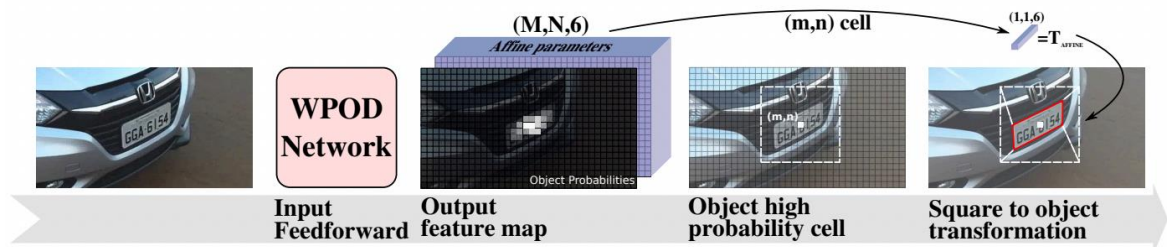
- Đối với tọa độ phía trên bên trái, ta kiểm tra và cập nhật giá trị như sau:
  - Nếu  $x_0 - \text{delta}_x < 0$  thì  $x_0 = 0$ , nếu không thì  $x_0 = x_0 - \text{delta}_x$
  - Nếu  $y_0 - \text{delta}_y < 0$  thì  $y_0 = 0$ , nếu không thì  $y_0 = y_0 - \text{delta}_y$
- Đối với tọa độ phía dưới bên phải, ta kiểm tra và cập nhật giá trị như sau:
  - Nếu  $x_0 + \text{delta}_x < w$  ( $w$  là chiều rộng của ảnh) thì  $x_0 = w$ , nếu không thì  $x_0 = x_0 + \text{delta}_x$
  - Nếu  $y_0 + \text{delta}_y < h$  ( $h$  là chiều cao của ảnh) thì  $y_0 = h$ , nếu không thì  $y_0 = y_0 + \text{delta}_y$

Sau khi tìm được tọa độ của mới của bounding box, ta cắt bounding box này ra khỏi hình ảnh để thu được hình ảnh chưa biển số xe với hằng số mở rộng ảnh  $k_x = 1.7$  và  $k_y = 2.9$ .

### 3.2.2. Xác định vùng chứa biển số sử dụng WPOD-NET

Theo như những gì tôi đã tham khảo được trong bài báo “*License Plate Detection and Recognition in Unconstrained Scenarios*” [11], một mạng tích chập đã được giới thiệu có tên là Warped Planar Object Detection Network (WPOD-Net) , được dịch sang tiếng việt là Mạng phát hiện đối tượng bị cong vênh. Mạng này học cách phát hiện biển số xe ở nhiều dạng biến dạng khác nhau và hội quy các hệ số của một phép biến đổi giúp “uốn nắn” biển số xe biến dạng thành hình chữ nhật giống như quan sát trực diện.

WPOD-NET được phát triển kế thừa những ưu điểm từ YOLO, SSD và Spatial Transformer Networks(STN) [14]. YOLO và SSD thực hiện phát hiện và nhận dạng nhiều đối tượng nhanh chóng cùng một lúc, nhưng chúng không tính đến các phép biến đổi không gian, chỉ tạo các hộp giới hạn hình chữ nhật cho mỗi lần phát hiện. Ngược lại, STN có thể được sử dụng để phát hiện các vùng không phải hình chữ nhật, tuy nhiên, nó không thể xử lý nhiều phép biến đổi cùng một lúc, chỉ thực hiện một phép biến đổi không gian duy nhất trên toàn bộ đầu vào.



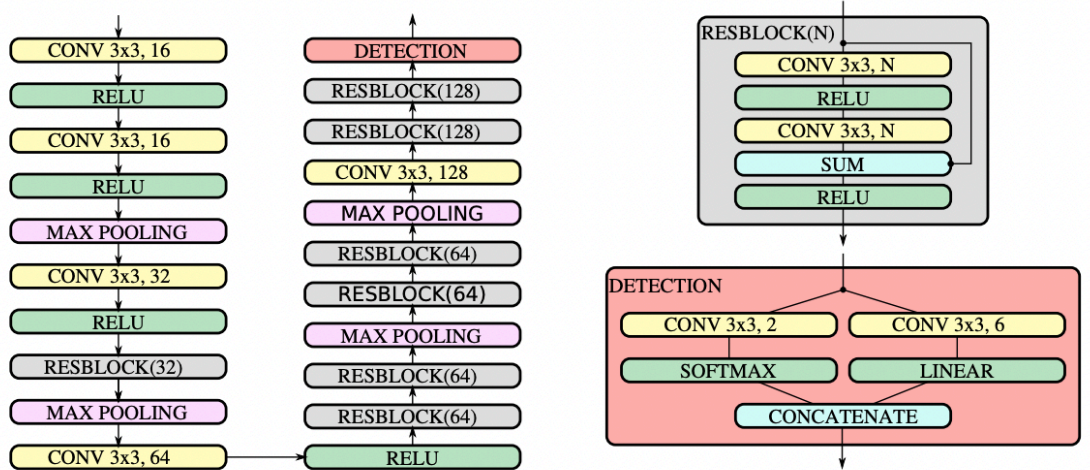
Hình 3.6 Hoạt động của mạng WPOD

Quá trình phát hiện bằng WPOD-NET được minh họa trong Hình 3. 8. Ban đầu, mạng được cung cấp bởi đầu ra của mô-đun phát hiện đối tượng là những phần vùng hình ảnh chứa biển số xe. Kết quả được lan truyền tiến (Feedforward) trong bản đồ đặc trưng (feature map) 8 kênh thứ mã hóa các xác suất đối tượng/không phải đối tượng và các tham số biến đổi affine. Để trích xuất biển số xe bị cong vênh, trước tiên hãy xem xét một vùng hình vuông tương tượng có kích thước cố định xung quanh tâm của một ô  $(m, n)$  . Nếu xác suất đối tượng cho ô này cao hơn ngưỡng phát hiện nhất định, thì một phần của



tham số hồi quy được sử dụng để xây dựng ma trận affine biến đổi vùng hình vuông tương ứng thành vùng biển số xe. Do đó, có thể dễ dàng uốn nắn biến số thành một đối tượng được căn chỉnh theo chiều ngang và chiều dọc.

**Kiến trúc mạng:** có tổng cộng 21 lớp tích chập, trong đó 14 lớp bên trong các khối dư (residual box) . Kích thước của tất cả các bộ lọc tích chập được cố định ở  $3 \times 3$ . Hàm kích hoạt ReLU được sử dụng trên toàn bộ mạng, ngoại trừ trong khối phát hiện. Có 4 lớp max pooling có kích thước  $2 \times 2$  với sai bằng 2 giúp giảm kích thước đầu vào xuống 16 lần. Cuối cùng, khối phát hiện có hai lớp tích chập song song: (i) một lớp để suy ra xác suất, được kích hoạt bởi hàm softmax và (ii) một cách khác để hồi quy các tham số affine mà không cần kích hoạt (hoặc, tương đương, sử dụng đơn vị  $F(x) = x$  làm hàm kích hoạt) .



Hình 3.7 Kiến trúc mạng WPOD

**Hàm mất mát:** Đặt  $p_i = [x_i, y_i]$ , với  $i = 1, 2, \dots, 4$ , biểu thị bốn góc của biển số được gắn nhãn, theo chiều kim đồng hồ bắt đầu từ điểm trên cùng bên trái. Ngoài ra đặt  $p_1 = [-0.5, -0.5]^T$ ,  $p_2 = [0.5, -0.5]^T$ ,  $p_3 = [0.5, 0.5]^T$ ,  $p_4 = [-0.5, 0.5]^T$  biểu thị các đỉnh tương ứng của một hình vuông đơn vị chính xác có tâm tại góc tọa độ.

Đối với hình ảnh đầu vào có chiều cao  $H$  và chiều rộng  $W$  và sai bước (stride) của mạng được cung cấp bởi  $N_s = 2^4$  (bốn lớp max pooling), feature map đầu ra của mạng bao gồm một volume  $M \times N \times 8$ , trong đó  $M = \frac{H}{N_s}$  và  $N =$



$\frac{W}{N_s}$ . Đối với mỗi ô điểm (m,n) trong feature map, có 8 giá trị được ước tính: hai giá trị đầu tiên (v1 và v2) là xác suất của đối tượng/không phải đối tượng và sau giá trị cuối cùng (v3 đến v8) được sử dụng để xây dựng phép biến đổi affine cục bộ  $T_{mn}$  được cho bởi:

$$T_{mn}(q) = \begin{bmatrix} \max(v3,0) & v4 \\ v5 & \max(v6,0) \end{bmatrix} q + \begin{bmatrix} v7 \\ v8 \end{bmatrix}$$

Trong đó, max function được sử dụng cho v3 và v6 để đảm bảo rằng đường chéo là dương (tránh phản chiếu không mong muốn hoặc xoay quá mức) .

Để phù hợp với độ phân giải đầu ra của mạng, các điểm  $p_i$  được thu nhỏ lại theo nghịch đảo của sai bước mạng và được căn giữa lại theo từng điểm (m,n) trong feature map. Điều này được thực hiện bằng cách áp dụng chức năng chuẩn hóa:

$$A_{mn}(p) = \frac{1}{\alpha} \left( \frac{1}{N_s} p - \begin{bmatrix} n \\ m \end{bmatrix} \right)$$

Trong đó,  $\alpha$  là hằng số tỷ lệ đại diện cho cạnh của hình vuông hư cấu. Nhóm tác giả đặt giá trị của  $\alpha = 7.75$ , là điểm trung bình giữa kích thước biên số tối đa và tối thiểu trong dữ liệu đào tạo tăng cường chia cho bước tiến của mạng. Giả sử rằng có một đối tượng(biến số) tại ô (m,n) , phần đầu tiên của hàm mất mát xem xét lỗi giữa phiên bản bị biến dạng của hình vuông chính tắc và các điểm được gán nhãn và chuẩn hóa của biên số xe, được cho bởi:

$$f_{affine}(m, n) = \sum_{i=1}^4 \|T_{mn}(q_i) - A_{mn}(p_i)\|_1$$

Phần thứ hai của hàm mất mát xử lý xác suất có/không có đối tượng tại (m,n) . Nó tương tự như độ tin cậy của SSD[21] và về cơ bản là tổng của hai hàm log-loss:

$$f_{probs}(m, n) = \text{logloss}(I_{obj}, v1) + \text{logloss}(1 - I_{obj}, v2)$$

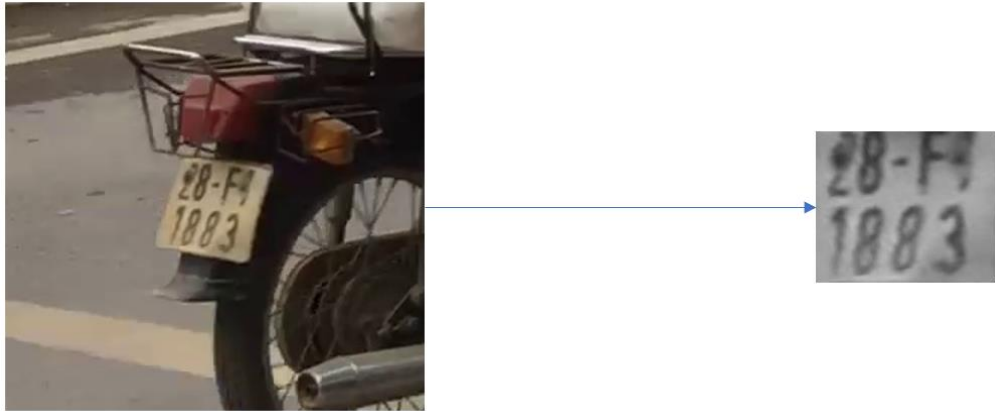
trong đó  $I_{obj}$  là hàm chỉ thị đối tượng trả về 1 nếu có một đối tượng tại điểm (m, n) hoặc 0 nếu ngược lại, và  $\text{logloss}(y, p) = -y \log(p)$  . Một đối tượng được xem xét bên trong một điểm (m, n) nếu hộp giới hạn hình chữ nhật của

nó thể hiện IoU lớn hơn ngưỡng  $\gamma_{obj}$  (được đặt theo kinh nghiệm là 0,3) w. r. t. một hộp giới hạn khác có cùng kích thước và được căn giữa tại  $(m, n)$ .

Hàm mất mát cuối cùng được đưa ra bởi sự kết hợp của cả hai thành phần, ta được:

$$loss = \sum_{m=1}^M \sum_{n=1}^N [II_{obj} f_{affine}(m, n) + f_{probs}(m, n)]$$

Dưới đây là ví dụ về kết quả thực hiện xác định vùng biên số:



Hình 3.8 Kết quả sau khi sử dụng WPOD-NET

### 3.2.3. Phân đoạn từng ký tự trên biển số

Ảnh đầu vào sẽ được chuyển màu từ BGR (Blue -Green- Red) về ảnh đa cấp xám nhằm cải thiện độ chính xác.

Adaptive threshold được áp dụng để làm nổi bật những kí tự, hoạt động như bộ lọc (filter) đi từ trái qua phải, trên xuống dưới và sẽ lấy những giá trị pixel nào lớn hơn ngưỡng (threshold) và một giá trị offset mà tôi điều chỉnh. Khác với các kiểu threshold truyền thống áp dụng ngưỡng cho toàn bộ ảnh, adaptive threshold áp dụng cho từng vùng nhỏ nhờ đó hoạt động tốt hơn với điều kiện ánh sáng kém.

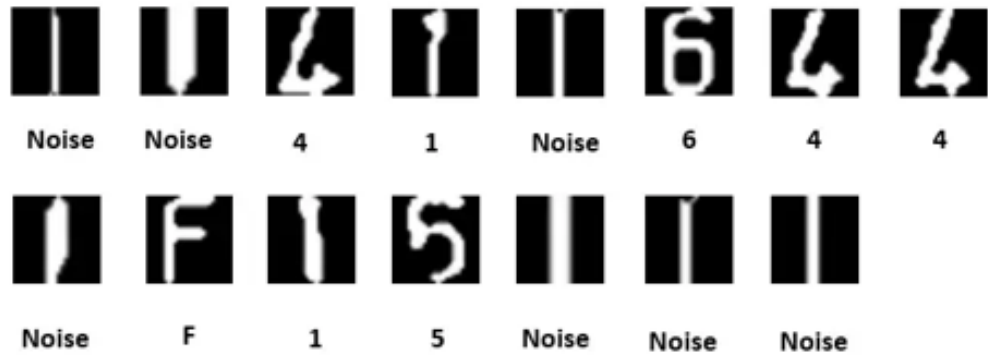
Một ví dụ khi qua bước xử lý này:



Hình 3.9 Ví dụ về kết quả sau khi áp dụng adaptive threshold

Để tách từng kí tự ra khỏi biển số xe, tôi áp dụng thuật toán Connected components analysis. Thuật toán này có ý tưởng đơn giản là nó sẽ kết nối tất cả pixel nào có cùng giá trị thành một khối và gán cho nó một cái nhãn (label). Nhờ đó tất cả các pixel của cùng một kí tự do có cùng giá trị sẽ được kết nối và được tách ra khỏi biển số xe. Tuy nhiên để tiện xử lý thì toàn bộ màu trắng được chuyển thành đen và ngược lại. Đồng thời cũng làm mờ ảnh để loại những nhiễu tròn vì có thể nó làm kí tự gắn liền với những bộ phận không mong muốn và thuật toán Connected components analysis (CCA) sẽ bị ảnh hưởng.

Sau khi thực hiện CCA, thu được một dãy các giá trị labels bao gồm các kí tự, các nhiễu không mong muốn (khung biển số xe, dấu gạch ngang, dấu chấm,...) và background (có label = 0 theo mặc định của hàm label). Do đó đầu tiên tôi loại bỏ các label của background sau đó áp dụng hàm findContours để tạo các cạnh bao quanh các kí tự cũng như các nhiễu. Do ở bước này các giá trị thu được ngoài kí tự còn có cả nhiễu do đó thiết lập các giá trị ngưỡng để loại bỏ nhiễu. Ở đây sử dụng ngưỡng đối với ba đại lượng: aspect ratio (tỉ lệ rộng / dài), solidity (tỉ lệ diện tích phần contour bao quanh kí tự và hình chữ nhật bao quanh kí tự) và height ratio (tỉ lệ chiều dài kí tự / chiều dài biển số xe). Nếu sử dụng phương pháp deep learning phân loại thì tốn thời gian bởi lượng nhiễu là rất lớn. Tôi chỉ cần tinh chỉnh những tham biến này tùy tập dữ liệu đã có thể nhanh chóng loại bỏ phần lớn nhiễu một cách đơn giản, lấy ra ảnh kí tự trên biển số xe mong muốn. Sau đó đưa về dạng (28, 28, 1) là kích thước input đầu vào của mạng CNN và cũng thêm tọa độ (x, y) của từng chữ cái.



Hình 3.10 Kết quả sau khi phân đoạn từng ký tự

#### 3.2.4. Phân loại ký tự sử dụng mô hình CNN

Ở Việt Nam, trên biển số xe chấp nhận 31 ký tự bao gồm cả chữ và chữ số (0-9). Tuy nhiên, ở kết quả mặc dù tôi đã sử dụng ngưỡng để loại bỏ nhiễu nhưng vẫn có một số nhiễu vẫn lọt qua vì nó có kích thước hình dáng đủ tiêu chuẩn như ký tự thường. Do có số lượng ít nên ở bước này thêm một class là background để dùng model CNN phân loại đâu là ký tự, đâu là nhiễu. Vậy tổng các class cần phân loại là 32.

Bộ dữ liệu để huấn luyện model ở đây gặp hai vấn đề. Thứ nhất là bộ dữ liệu có sẵn như MNIST thì quá đẹp nhưng khi vào môi trường thực tế với điều kiện ánh sáng yếu thì ký tự thu được sẽ bị méo mó chút như hình 3.6. Và vấn đề thứ hai là font chữ nước ngoài khác font chữ Việt Nam. Hai yếu tố này sẽ tác động đến chất lượng của mô hình phân loại (model classification). Do đó tác giả bài báo [16] đã tự xây dựng bộ datasets về biển số xe Việt Nam để đảm bảo chất lượng mô hình.

Sau đó tác giả xây dựng một mô hình CNN đơn giản để tiến hành trích xuất đặc trưng và phân loại. Mô hình này được thiết kế với các số bộ lọc tăng dần [32, 64, 64] để sao cho càng gần về đầu ra không gian đặc trưng giảm dần nhưng số lượng cách học (= số bộ lọc) tăng dần. Tùy độ phức tạp của dữ liệu mà tôi có thể thay đổi. Cuối cùng sử dụng một lớp làm phẳng và một activation softmax để phân loại. Giá trị 32 ở đây chính là số class đề cập bên trên.

### 3.2.5. Định dạng lại biến số

Ở bước cuối cùng này, tôi sử dụng tọa độ  $(x, y)$  trên mỗi kí tự đã đề cập. Mặt phẳng tọa độ  $xOy$  trong đó  $x$  là trục ngang,  $y$  là trục dọc. Do biến số xe đã được đưa về hướng chính diện nên có thể xác định biến số xe đang xét là một dòng hay hai dòng dựa vào tọa độ của các kí tự. Ta sẽ đặt ra một ngưỡng để xét xem biến số là một dòng hay hai dòng. Nếu là biến số xe một dòng thì tọa độ  $y$  của tất cả kí tự thu được sẽ nhỏ hơn ngưỡng. Còn nếu là biến số hai dòng thì tọa độ  $y$  chênh lệch giữa các kí tự sẽ lớn hơn ngưỡng. Sau đó ta sẽ sử dụng tọa độ  $x$  để sắp xếp các kí tự theo quy luật từ trái qua phải từ trên xuống dưới. Khi đó, ta thu được biến số xe cần trích xuất.

## CHƯƠNG 4: THỰC NGHIỆM

### 4.1. Chuẩn bị bộ dữ liệu

#### 4.1.1. Bộ dữ liệu phát hiện biển số xe

Bộ dữ liệu phát hiện biển số xe được thu thập từ video ghi lại tại một đoạn đường tại Chợ Cống, Chương Mỹ, Hà Nội. Video ghi lại có chất lượng 1080p với tốc độ 30 khung hình trên giây. Từ 1 video thô dài 20 phút đã được ghi lại, tôi sẽ lấy 3 ảnh/giây tổng cộng 3600 ảnh có thể được sử dụng để tiến hành gán nhãn. Thiết bị được sử dụng ở đây là Apple iphone 6s được đặt cách mặt đường 0.5 m. Sau đó tiến hành gán nhãn toàn bộ khung hình trong video xử lý.



Hình 4.1 Ảnh ví dụ tại điểm khảo sát

Bộ dữ liệu 930 ảnh được chia theo tỉ lệ 88%, 8%, 4% trong đó 88% là dữ liệu ảnh cho tập huấn luyện, 8% ảnh dành cho tập đánh giá, và 4% dành cho tập test. Để tăng tính đa dạng cho bộ dữ liệu, tôi sử dụng thêm các kĩ thuật làm giàu dữ liệu để tăng cường bộ dữ liệu bằng cách biến đổi các ảnh trong bộ dữ liệu mà không cần thêm ảnh mới: xoay ảnh, làm biến dạng ảnh, điều chỉnh độ tương phản, làm mờ.

#### 4.1.2. Bộ dữ liệu nhận diện biển số xe

Để huấn luyện cho mô hình nhận dạng biển số xe, tôi đã nghiên cứu và lựa chọn bộ ảnh biển số xe máy của công ty GreenParking, đây là bộ dữ liệu chứa đựng biển số xe tại Việt Nam. Tôi đã lấy ngẫu nhiên 265 bức ảnh khác nhau từ bộ ảnh và tự gán nhãn đồng thời lưu vào file labels.csv.

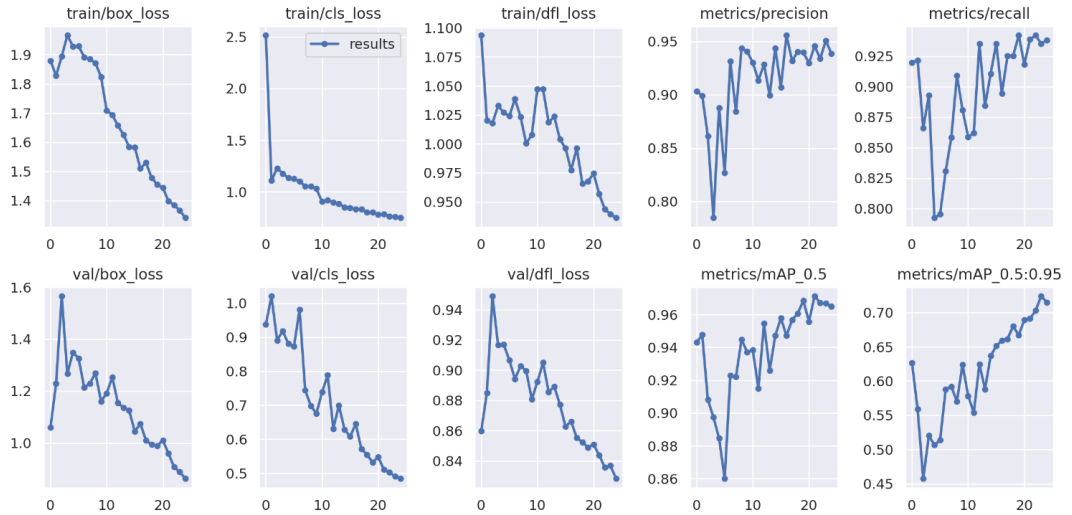


Hình 4.2 Một vài hình ảnh trong bộ dữ liệu của công ty GreenParking

#### 4.2. Huấn luyện mô hình nhận diện vật thể

Thuật toán YOLOv9 được sử dụng để huấn luyện mô hình phát hiện đối tượng trên Google Collaboratory. Google Collaboratory cung cấp môi trường biên dịch ngôn ngữ Python chạy trên nền tảng web và sử dụng các phần cứng như Tesla T4 GPU với 15GB RAM. Mô hình nhận diện biển số xe được huấn luyện trên mô hình YOLOv9c đã được huấn luyện với 25 epochs và batch size là 16. Hình ảnh bên dưới là kết quả của việc huấn luyện:





Hình 4.3 Kết quả huấn luyện mô hình phát hiện vật thể

### 4.3. Đánh giá trên tập thực nghiệm

#### 4.3.1. Mô hình phát hiện biển số xe

Mô hình phát hiện đối tượng YOLOv9c được lựa chọn đem lại độ chính xác cao khi đánh giá trên dữ liệu thử nghiệm (test data) với các con số ấn tượng như: mAP = 96,7%, Precision = 95,1% và Recall = 93,5%.

Để kiểm tra khả năng của hệ thống, tôi đã thử nghiệm hệ thống trên các điều kiện thực tế. Dưới đây là một vài kết quả thực nghiệm:



Hình 4.4 Một số hình ảnh kết quả của mô hình phát hiện biển số xe



Về ưu điểm:

- Mô hình YOLOv9 nhanh và chính xác.
- Mô hình này thể hiện tốt với nhiễu và có thể khái quát trên dữ liệu mới.

Nhược điểm:

- Mô hình kích thước lớn đem lại độ chính xác cao nhưng số lượng tham số quá lớn khiến khó có thể triển khai trên thiết bị biên.

#### 4.3.2. Mô hình nhận dạng biển số xe

Tôi sẽ kiểm tra bằng cách thực hiện mô hình nhận dạng biển số xe cho từng ảnh sau đó so sánh kết quả nhận dạng với nhãn đã gán nếu trùng khớp sẽ được tính là chính xác và ngược lại. Và tôi có bảng kết quả:

**Bảng 4 Kết quả nhận dạng biển số xe trên bộ dữ liệu GreenParking**

	Số ảnh nhận dạng	Tỉ lệ
Nhận dạng chính xác	162	61%
Nhận dạng đoán sai	103	39%

Dựa trên bảng trên, tổng số ảnh được nhận dạng thành công là 265/265 (tổng số ảnh trong bộ dữ liệu) với 162 lần nhận dạng chính xác chiếm tỉ lệ 61% và 103 lần nhận dạng chưa chính xác chiếm tỉ lệ 39%.

Một số hình ảnh về kết quả sau thực hiện mô hình nhận dạng biển số xe:



**Hình 4.5 Một số hình ảnh kết quả của mô hình nhận dạng biển số xe**

Về ưu điểm:

- Tất cả biển số đều được phát hiện
- Mô hình có kích thước khá nhẹ nên tốc độ xử lý nhanh
- Nhận dạng tương đối chính xác, ngay cả khi biển số xe bị che khuất một phần nào nhỏ

Về nhược điểm:

- Khi ảnh đầu vào bị nghiêng, một vài kí tự sẽ bị nhận dạng không đúng dòng.
- Hoạt động không chính xác khi ảnh đầu vào mờ.
- Đôi khi phân loại sai giữa số 8 và chữ B, số 0 và chữ D, chữ Y và chữ V.

#### 4.4. Triển khai hệ thống

Sau khi đã hoàn thành huấn luyện và thử nghiệm trên tập dữ liệu thực tế, các mô hình được ghép lại thành hệ thống nhận diện biển số xe.

Thiết bị sử dụng: Laptop MSI bravo 15, sử dụng chip AMD Ryzen 4600H, 16GB RAM.

Mô trường yêu cầu có: keras 2.15, tensorflow 2.15, torch, opencv python, numpy,...

Đầu vào: video ghi lại cảnh đường phố

Đầu ra: thông tin biển số xe của các phương tiện tham gia giao thông.



Hình 4.6 Một số hình ảnh kết quả của việc triển khai mô hình

## KẾT LUẬN

Qua quá trình học hỏi, nghiên cứu, tìm hiểu những nghiên cứu đi trước trong lĩnh vực thị giác máy tính về nhận diện vật thể (Object detection) và nhận dạng ký tự quang học (Optical Character Recognition) đã giúp tôi hoàn thành đề tài “*Xây dựng mô hình nhận diện biển số xe bằng WPOD*”.

Tôi đã tìm hiểu, nghiên cứu và cài đặt thử nghiệm các thuật toán, các mô hình mạng trí tuệ nhân tạo liên quan đến bài toán nhận diện vật thể như mô hình phát hiện biển số xe, mô hình nhận diện ký tự quang trong biển số xe. Ngoài ra, bên cạnh những tập dữ liệu có sẵn, tôi cũng đã cố gắng thu thập những dữ liệu mới phục vụ cho bài toán này.

Hệ thống cho độ chính xác khá là tốt. Đối với mô hình nhận diện vật thể, mAP đạt 96,7% dù cho số lượng epoch chỉ có là 25. Còn đối với mô hình nhận dạng biển số xe trong tổng số ảnh 265 từ bộ dữ liệu GreenParking thì đã nhận diện thành công 162 ảnh chiếm tỉ lệ là 61%.

Tôi xin chân thành cảm ơn thạc sĩ Nguyễn Thị Hương Lan đã hỗ trợ tôi giúp tôi hoàn thành đề tài này. Đề tài này đã giúp tôi củng cố, học hỏi và rèn luyện rất nhiều kỹ năng trong việc thu thập thông tin, tài liệu hữu ích cho việc ứng dụng những kiến thức đã thu thập được vào bài toán thực tế.

Tuy nhiên do thời gian và tài nguyên có hạn, nên hệ thống hiện tại vẫn còn rất nhiều thiếu sót như độ chính xác của mô hình nhận dạng ký tự vẫn chưa được thực sự tốt. Hệ thống vẫn chưa có giao diện tính chỉnh và sử dụng dễ dàng. Vì vậy tôi rất mong được các thầy cô đóng góp thêm các ý kiến để có thể cải thiện hệ thống này.

Tôi xin trân trọng cảm ơn!

## TÀI LIỆU THAM KHẢO

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2015, “You Only Look Once: Unified, Real-Time Object Detection”.
- [2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , Honolulu, HI, USA, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.
- [3] Redmon, Joseph and Ali Farhadi (2018) . “YOLOv3: An Incremental Improvement. ”
- [4] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan, (2020) , “YOLOv4: Optimal Speed and Accuracy of Object Detection”.
- [5] Ultralytics, 2020, "Ultralytics YOLOv5"
- [6] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei, 2022, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications”
- [7] Wang, Chien-Yao & Bochkovskiy, Alexey & Liao, Hong-yuan. (2022) . YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.
- [8] Ultralytics, 2023, "Ultralytics YOLOv8"
- [9] Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao, 2024, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”
- [10] Matthew Hill, 2014, "OpenALPR"

- [11] S. Montazzolli Silva, C. Rosito Jung, 2018, "License Plate Detection and Recognition in Unconstrained Scenarios".
- [12] Du S. , Ibrahim M. , Shehata M. , Badawy W. , “Automatic license plate recognition (ALPR): A state-of-the-art review”.
- [13] Zheng D. , Zhao Y. , Wang J. , 26 (15) (2005) , An efficient method of license plate location Pattern Recognit. Lett. , pp. 2431-2438
- [14] Du S. , Ibrahim M. , Shehata M. , Badawy W. , 23 (2) (2013) , Automatic license plate recognition (ALPR): A state-of-the-art review IEEE Trans. Circuits Syst. Video Technol. , pp. 311-325
- [15] Xifan Shi, Weizhong Zhao, Yonghang Shen (2005) , Automatic License Plate Recognition System Based on Color Image Processing
- [16] Bùi Quang Mạnh, 2020, “Nhận diện biển số xe Việt Nam”