

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

Môn học: CS331.O11.KHCL

**ĐỀ TÀI: Nhận dạng và đọc biển số xe**

**Giảng viên môn học:** TS . Mai Tiến Dũng

**Sinh viên thực hiện:**

Trần Duy Thanh- 20521925

Nguyễn Anh Tuấn-20522114

TP. Hồ Chí Minh, ngày 8 tháng 1 năm 2024

## MỤC LỤC

1	GIỚI THIỆU .....	1
1.1	Giới thiệu đề tài .....	1
1.2	Xác định bài toán .....	1
1.3	Các bước thực hiện .....	1
1.4	Mô tả và xây dựng bộ dữ liệu .....	1
1.5	Thống kê dữ liệu .....	2
2	THUẬT TOÁN .....	3
2.1	Lý do chọn thuật toán .....	3
2.2	Giới thiệu thuật toán .....	3
2.3	Giới thiệu YOLOV5 .....	4
3	QUÁ TRÌNH HUẤN LUYỆN VÀ THỰC NGHIỆM .....	6
3.1	Quá trình huấn luyện .....	6
3.2	Các thông số đánh giá .....	7
3.3	Kết quả .....	8
3.4	Đánh giá và nhận xét .....	12
3.5	Kết hợp 2 mô hình .....	13
4.	TỔNG KẾT .....	17



# 1 GIỚI THIỆU

## 1.1 Giới thiệu đề tài

Trong lĩnh vực giám sát giao thông, an ninh, và quản lý đô thị, các hệ thống sử dụng học máy và xử lý ảnh để tự động nhận diện và giải mã thông tin từ biển số xe. Điều này giúp cải thiện khả năng quản lý và kiểm soát phương tiện giao thông, đồng thời đáp ứng nhanh chóng các thách thức trong xã hội hiện đại.

Mục tiêu tạo ra một hệ thống tự động có khả năng nhận diện và đọc biển số xe. Giúp cải thiện hiệu suất trong việc tự động hóa quá trình nhận diện phương tiện giao thông và thu thập thông tin liên quan.

## 1.2 Xác định bài toán

Input: Ảnh (.png và .jpg) chứa các thông tin của các phương tiện giao thông di chuyển trên đường phố Việt Nam.

Output: Ảnh, trong đó các biển số xe đã được phát hiện và đọc biển số xe.

## 1.3 Các bước thực hiện

Để thực hiện đề án, nhóm đã thực hiện các bước như sau:

- **Thu thập dữ liệu:** Nhóm đã thu thập một số lượng lớn hình ảnh và video chứa các biển số xe để huấn luyện và kiểm thử hệ thống.
- **Chuẩn bị dữ liệu:** Dữ liệu thu thập được được chia thành các tập dữ liệu để huấn luyện và kiểm thử mô hình.
- **Huấn luyện mô hình:** Sử dụng một mô hình mạng nơ-ron tích chập để huấn luyện cho nhiệm vụ nhận diện biển số xe trên hình ảnh hoặc video.
- **Kiểm thử và tinh chỉnh:** Sau khi huấn luyện, tiến hành kiểm thử và tinh chỉnh mô hình để đạt được độ chính xác cao nhất.
- **Đọc biển số xe**

## 1.4 Mô tả và xây dựng bộ dữ liệu

Mô tả nguồn dữ liệu

### \*Tự thu thập dữ liệu

Video tự quay: 10 video, trung bình mỗi video có độ dài 2 phút. Địa

điểm : Quốc lộ 1K, cầu vượt nằm trên Xa lộ Hà Nội.



## Cách thu thập:

Sử dụng điện thoại cá nhân quay với lượng video 720p

Khi tiến hành đi quay, giữ máy quay cố định để có thể lấy được dữ liệu biển số trên tuyến đường.

Nhóm tiến hành quay vào lúc trời sáng trưa, chiều và gần tối.

### \*Dữ liệu trên mạng

Lấy từ mì AI: Hình ảnh lấy trên Mì AI (216 tấm ảnh)

### Xử lý dữ liệu

Sau khi có được dữ liệu từ video, nhóm tiến hành cắt ghép ảnh từ video với 0.5 frame/second- tương đương với 1 giây là 2 hình ảnh. Sau đó các hình ảnh bị mờ không thấy rõ biển số sẽ được xóa bỏ. Và giữ lại những tấm thấy được biển số để gán nhãn.

### Gán nhãn dữ liệu

Nhóm sử dụng Roboflow - là một framework dành cho nhà phát triển Thị giác Máy tính để thu thập dữ liệu tốt hơn, xử lý trước và các kỹ thuật đào tạo mô hình .

Đường dẫn truy cập: <https://roboflow.com>

## 1.5 Thống kê dữ liệu

Nhóm sử dụng 2 bộ dữ liệu để có thể nhận diện biển số xe bao gồm bộ dữ liệu về biển số xe và bộ dữ liệu về ký tự trên biển số.

**Bộ dữ liệu License-Plate** cho biết biển số xe trong ảnh, trong đó gồm 3823 ảnh chia vào các tập train, test, valid với tỷ lệ 70,20,10 với 2883 ảnh trong tập train, 627 ảnh trong tập valid và 313 ảnh trong tập test. Dữ liệu gồm 4 loại biển khác nhau bao gồm biển trắng, biển xanh, biển đỏ, biển vàng.

Đường dẫn truy cập: <https://universe.roboflow.com/uit-5rhkt/lisence-car>

**Bộ dữ liệu Characters-Plate** cho biết thông tin biển số xe trong ảnh, gồm 6026 ảnh với 4224 ảnh trong tập train, 1200 trong tập valid và 602 trong tập test. Dữ liệu gồm 32 class khác nhau tương ứng với chữ và số có trong biển số Việt Nam.

Đường dẫn truy cập: <https://universe.roboflow.com/license-plate-bpxif/number-plate>

## 2 THUẬT TOÁN

### 2.1 Lý do chọn thuật toán

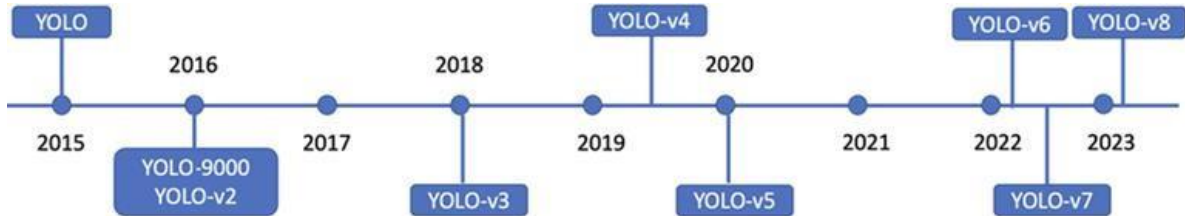
Trong việc lựa chọn thuật toán, nhóm em đã xem xét tiêu chí quan trọng:

- Độ chính xác
- Tốc độ xử lý
- Hỗ trợ nhiều tác vụ
- Khái quát tốt

YOLO là một thuật toán phù hợp đáp ứng những tiêu chí trên.

### 2.2 Giới thiệu thuật toán

YOLO (You Only Look Once) là mô hình phát hiện đối tượng phổ biến được biết đến với tốc độ nhanh và độ chính xác cao. Mô hình này lần đầu tiên được giới thiệu bởi Joseph Redmon và cộng sự vào năm 2016. Kể từ đó đến nay, đã có nhiều phiên bản của YOLO, một trong những phiên bản gần đây nhất là YOLO v8.

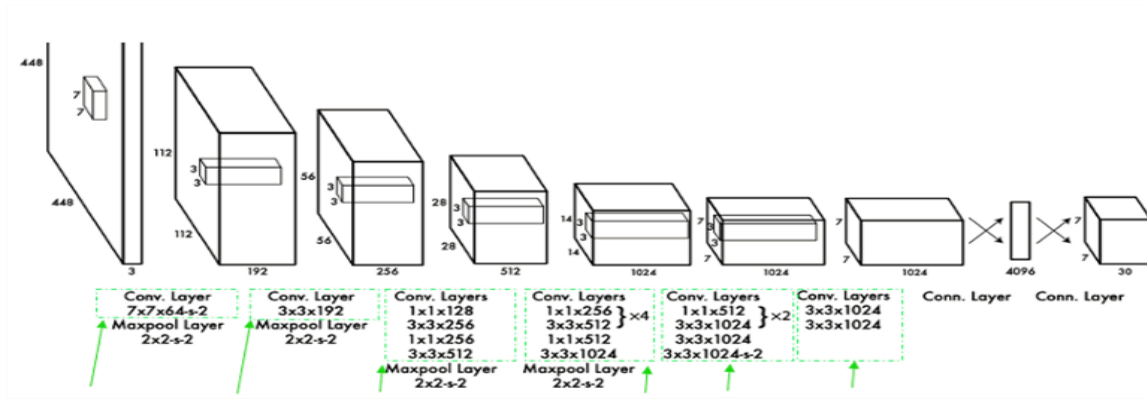


Hình 1: Yolo

### Kiến trúc YOLO

Kiến trúc YOLO tương tự như [GoogleNet](#), nó có tổng thể 24 lớp tích chập, bốn lớp tổng hợp tối đa và hai lớp được kết nối đầy đủ.

Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



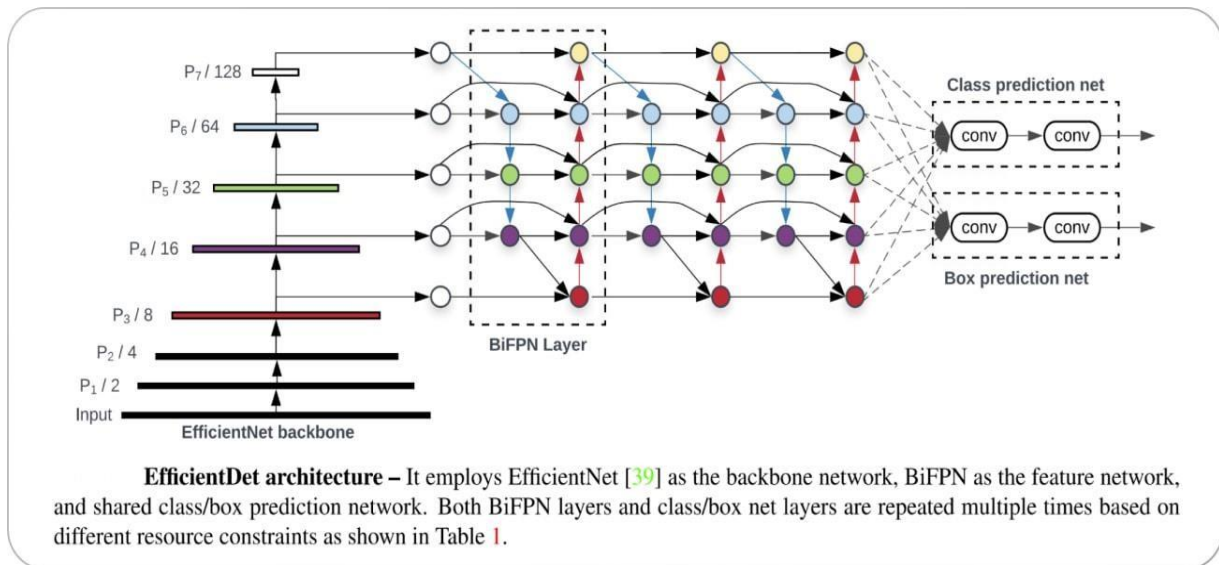
Hình 2: Kiến trúc Yolo

### 2.3 Giới thiệu YOLOV5

YOLOv5 được giới thiệu vào năm 2020 bởi cùng một nhóm đã phát triển thuật toán YOLO ban đầu dưới dạng một dự án mã nguồn mở và được duy trì bởi Ultralytics.

YOLOv5 được xây dựng dựa trên sự thành công của các phiên bản trước và bổ sung một số tính năng và cải tiến mới.

YOLOv5 sử dụng một kiến trúc phức tạp hơn gọi là EfficientDet (kiến trúc hiển thị bên dưới), dựa trên kiến trúc mạng EfficientNet. Việc sử dụng một kiến trúc phức tạp hơn trong YOLOv5 cho phép nó đạt được độ chính xác cao hơn và khả năng khái quát hóa tốt hơn cho nhiều loại đối tượng hơn.



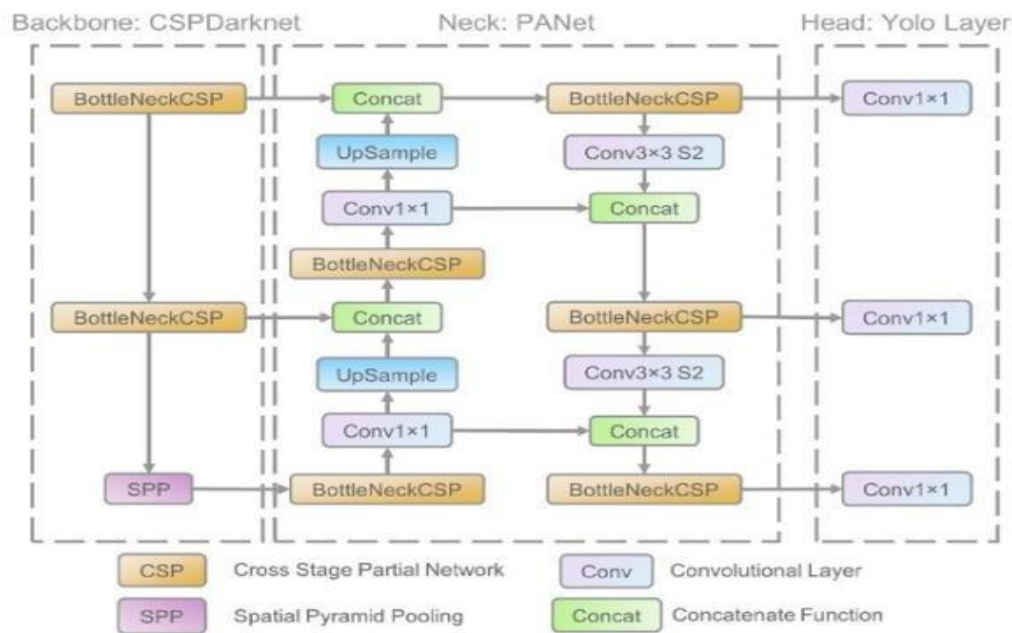
Hình 3: Kiến trúc YoloV5

YOLO và YOLO v5 khác biệt trong việc sử dụng dữ liệu huấn luyện. YOLO được huấn luyện trên bộ dữ liệu PASCAL VOC (20 loại đối tượng), trong khi YOLO v5 sử dụng bộ dữ liệu D5 (600 loại đối tượng).

Áp dụng "**dynamic anchor boxes**" để tạo hộp neo dựa trên cụm hộp thực tế, giúp chúng chính xác hơn với kích thước và hình dạng của đối tượng.

YOLO v5 cũng sử dụng "**spatial pyramid pooling**" để cải thiện phát hiện đối tượng nhỏ và giới thiệu "**CIoU loss**" để tăng cường hiệu suất trên bộ dữ liệu không cân đối.

### Kiến trúc mạng cho YOLO v5



Hình 4: Kiến trúc mạng cho YOLO v5

### Kiến trúc của YOLOv5 gồm ba phần chính:

*CSP-Darknet53 (Cross Stage Partial-Darknet53):*

Là một mạng cơ bản cho YOLOv5.

Sử dụng chiến lược CSPNet để giảm redundant gradients và tăng tốc mô hình.

*Spatial Pyramid Pooling (SPP):*

Tăng kích thước receptive field và tách biệt đặc trưng quan trọng từ đầu vào. Sử dụng biến thể SPPF để tăng tốc mô hình.

*Path Aggregation Network (PANet):*

Cải thiện thông tin và giúp dự đoán chính xác vị trí pixel trong nhiệm vụ dự đoán mask. Được cải tiến bằng cách áp dụng chiến lược CSPNet.

*Head of the Network (Đầu Mạng):*

Dự đoán tọa độ và loại của đối tượng. Giống với YOLOv3 và YOLOv4.

### 3 QUÁ TRÌNH HUẤN LUYỆN VÀ THỰC NGHIỆM

Để train mô hình nhóm thiết lập các trọng số sau:

--img 640: Kích thước ảnh đầu vào được sử dụng trong quá trình huấn luyện.

--batch 16: Số lượng ảnh được sử dụng trong mỗi batch huấn luyện.

--epochs 100: Số lượng epochs (vòng lặp huấn luyện) để huấn luyện mô hình.

#### 3.1 Quá trình huấn luyện

Huấn luyện mô hình YOLOv5

Nhóm tiến hành sử dụng phiên bản YOLOv5 m vì nó có độ chính xác cao, thời gian chạy tốt và tài nguyên sử dụng trung bình.

Model	size (pixels)	mAP <sup>val</sup> 0.5:0.95	mAP <sup>val</sup> 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

Hình 5: Các loại mô hình YoloV5

**Bộ dữ liệu License-Plate**

Mô hình mất **2 tiếng** để hoàn thành việc học trên tập train.

```
Model summary: 212 layers, 20869098 parameters, 0 gradients, 47.9 GFLOPs
```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	627	761	0.928	0.977	0.978	0.725
Bien_do	627	24	0.948	1	0.985	0.714
Bien_ngoaigiao	627	17	0.956	1	0.995	0.672
Bien_trang	627	652	0.903	0.974	0.982	0.809
Bien_vang	627	45	0.876	0.956	0.944	0.675
Bien_xanh	627	23	0.956	0.956	0.986	0.756

Results saved to runs/train/exp

Hình 6: Kết quả sau khi train phát biển biển số xe với 100 epoch





## Bộ dữ liệu Characters-Plate

Mô hình mất 4 tiếng để hoàn thành việc học trên tập train.

Model summary: 212 layers, 20994369 parameters, 0 gradients, 48.3 GFLOPs							
Class	Images	Instances	P	R	mAP50	mAP50-95	100% 38/38 [00:22<00:00, 1.72it/s]
all	1200	9968	0.972	0.95	0.978	0.724	
0	1200	752	0.985	0.97	0.977	0.76	
1	1200	1226	0.99	0.965	0.991	0.647	
2	1200	829	0.995	0.98	0.995	0.761	
3	1200	627	0.976	0.971	0.984	0.741	
4	1200	722	0.993	0.979	0.992	0.73	
5	1200	1099	0.991	0.993	0.992	0.782	
6	1200	962	0.991	0.984	0.991	0.762	
7	1200	850	0.994	0.987	0.993	0.743	
8	1200	592	0.985	0.974	0.985	0.743	
9	1200	943	0.99	0.992	0.989	0.749	
A	1200	100	1	0.981	0.994	0.743	
B	1200	223	0.977	0.973	0.988	0.799	
C	1200	74	0.943	0.986	0.969	0.716	
D	1200	76	0.99	0.921	0.974	0.708	
E	1200	49	1	0.935	0.989	0.716	
F	1200	109	0.991	0.997	0.988	0.727	
G	1200	70	0.986	0.999	0.995	0.768	
H	1200	57	0.894	0.895	0.96	0.704	
I	1200	2	0.966	1	0.995	0.622	
J	1200	6	1	0.69	0.922	0.659	
K	1200	57	0.98	0.93	0.951	0.678	
L	1200	145	0.986	1	0.99	0.814	
M	1200	39	0.838	0.821	0.944	0.671	
N	1200	37	0.968	0.818	0.923	0.665	
P	1200	51	0.998	0.98	0.989	0.698	
R	1200	11	0.845	1	0.988	0.775	
S	1200	63	0.984	0.985	0.995	0.742	
T	1200	40	0.993	0.95	0.993	0.771	
U	1200	24	0.954	0.917	0.986	0.717	
V	1200	27	0.927	1	0.952	0.681	
X	1200	41	1	0.969	0.995	0.667	
Y	1200	12	0.96	0.833	0.917	0.691	
Z	1200	53	1	0.984	0.995	0.757	

Results saved to runs/train/exp

Hình 7: Kết quả sau khi train phát hiện ký tự biển số xe với 100 epoch

### 3.2 Các thông số đánh giá

Các thông số đánh giá chung bao gồm

**Precision (Chính xác):** Tỷ lệ giữa số lượng các đối tượng được dự đoán đúng so với tổng số lượng đối tượng được dự đoán. Nó đo lường khả năng của mô hình đưa ra dự đoán chính xác.

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

**Recall (Nhớ lại):** Tỷ lệ giữa số lượng đối tượng được dự đoán đúng so với tổng số lượng đối tượng thực tế. Nó đánh giá khả năng của mô hình nhận diện toàn bộ các đối tượng thực tế.

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

F1 Score: Là sự kết hợp của Precision và Recall. Nó cung cấp một đánh giá tổng thể về hiệu suất của mô hình, ưu tiên mô hình có cả hai chính xác và độ nhạy cao.

$$F1_{score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Intersection over Union (IoU): Đo lường độ chồng lấp giữa bounding box dự đoán và bounding box thực tế. Nó giúp đánh giá chất lượng của việc định vị đối tượng.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Ground truth box} \cup \text{Detected box}}$$

Hình 8: Công thức tính IoU

Average Precision (AP): Đo lường chất lượng của mô hình trên nhiều ngưỡng độ tin cậy khác nhau. Área dưới đường Precision-Recall thể hiện độ chính xác của mô hình trên toàn bộ phạm vi.

Mean Average Precision (mAP): Là giá trị trung bình của Average Precision trên tất cả các lớp đối tượng. Nó đánh giá hiệu suất tổng thể của mô hình trên nhiều loại đối tượng.

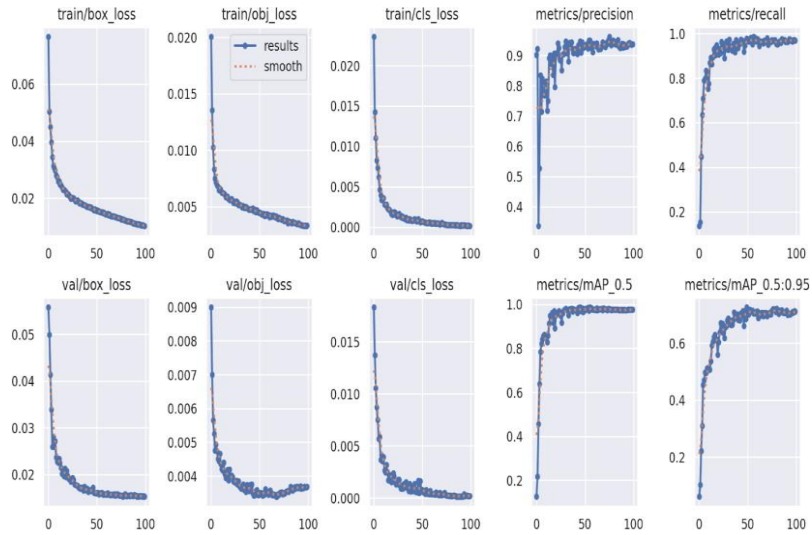
### 3.3 Kết quả

#### 1. Bộ dữ liệu License-Plate

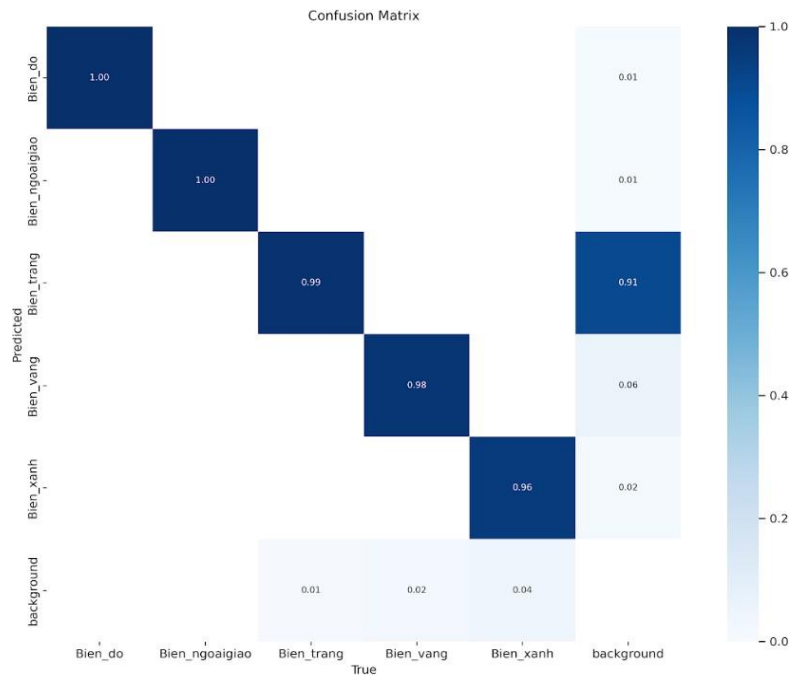
Kết quả train tập dữ liệu License-Plate sau 100 epoch:

Precision	Recall	mAP@.5	mAP@.5:.95
92.8%	97.7%	97.8%	72.5%

Bảng 1: Kết quả train tập dữ liệu License-Plate sau 100 epoch



Hình 9: Biểu đồ kết quả quá trình train 100 epochs



Hình 10: Confusion matrix với bộ dữ liệu License-Plate

## Kết quả dựa trên độ đo

Nhóm sử dụng IOU để xác định xem một bounding box dự đoán có chính xác định vị biển số xe hay không. Khi xác định một biển số xe, mô hình sẽ dự đoán một bounding box xung quanh biển số đó.

Sử dụng IOU với 0.5 để đánh giá hiệu suất của mô hình trên tập dữ liệu test

Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 10/10 [00:14:00:00, 1.41s/it]
all	313	369	0.962	0.954	0.978	0.744
Bien_do	313	11	1	0.898	0.988	0.781
Bien ngoaigiao	313	8	1	0.988	0.995	0.736
Bien_trang	313	307	0.924	0.984	0.986	0.833
Bien_vang	313	28	0.922	0.964	0.961	0.66
Bien_xanh	313	15	0.966	0.933	0.962	0.712

Hình 11: Hiệu năng của mô hình

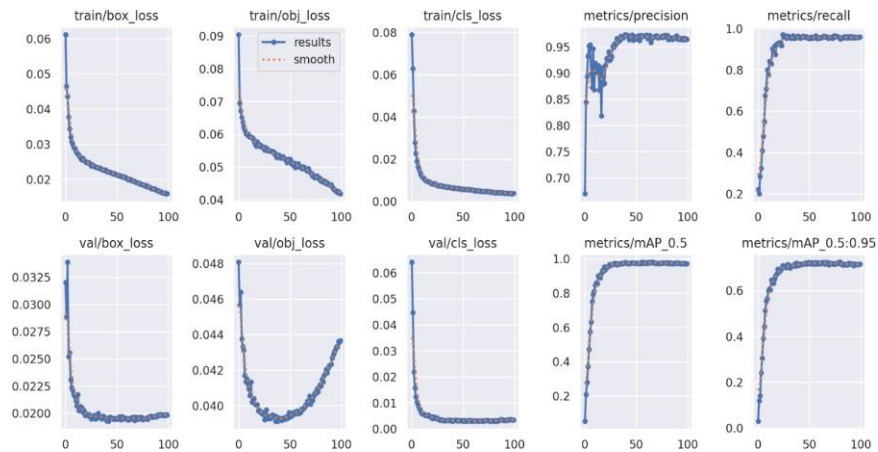
## 2. Bộ dữ liệu Character Plate

Kết quả train tập dữ liệu Character-Plate sau 100 epoch:

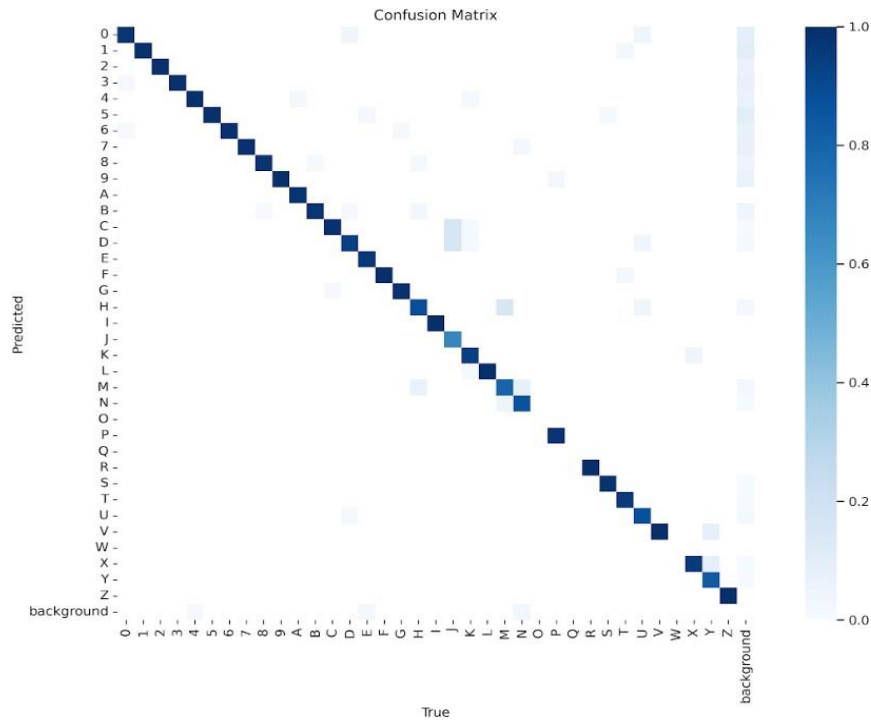
Precision	Recall	mAP@.5	mAP@.5:.95
97.2%	95%	97.8%	72.4%

Bảng 2: Kết quả train tập dữ liệu Character-Plate sau 100 epoch:

Thông số chi tiết kết quả training



Hình 12: Hình ảnh biểu đồ kết quả quá trình train 100 epochs



Hình 13: Confusion matrix với bộ dữ liệu License-Plate

## Kết quả dựa trên độ đo

Sử dụng IOU với 0.5 để đánh giá hiệu suất của mô hình trên tập dữ liệu test

Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 19/19 [08:58<00:00, 28.37s/it]
all	602	5016	0.958	0.965	0.974	0.73
0	602	391	0.992	0.964	0.983	0.75
1	602	636	0.994	0.972	0.993	0.667
2	602	386	0.997	0.988	0.994	0.751
3	602	330	0.979	0.989	0.988	0.757
4	602	348	0.991	0.986	0.99	0.739
5	602	558	0.996	0.987	0.992	0.771
6	602	468	0.985	1	0.992	0.771
7	602	425	0.988	0.995	0.99	0.746
8	602	283	0.992	0.989	0.989	0.74
9	602	461	0.993	0.988	0.993	0.762
A	602	66	0.998	1	0.995	0.726
B	602	123	1	0.992	0.995	0.795
C	602	47	1	0.944	0.956	0.717
D	602	48	0.979	0.978	0.992	0.735
E	602	25	0.996	1	0.995	0.806
F	602	42	0.928	1	0.988	0.741
G	602	43	0.997	1	0.995	0.763
H	602	27	0.918	0.926	0.984	0.698
I	602	3	0.975	1	0.995	0.73
J	602	6	0.955	0.833	0.835	0.573
K	602	41	0.974	0.908	0.951	0.703
L	602	80	0.998	1	0.995	0.777
M	602	19	0.954	0.842	0.959	0.727
N	602	17	0.854	0.941	0.913	0.709
P	602	25	0.994	1	0.995	0.774
R	602	7	0.975	1	0.995	0.801
S	602	19	0.992	1	0.995	0.71
T	602	19	0.989	1	0.995	0.731
U	602	7	0.758	1	0.964	0.79
V	602	13	0.986	0.846	0.975	0.7
X	602	10	0.815	1	0.889	0.616
Y	602	5	0.679	0.854	0.938	0.598
Z	602	38	0.996	0.921	0.946	0.716

Hình 14: Bảng đánh giá hiệu suất của mô hình trên tập dữ liệu test

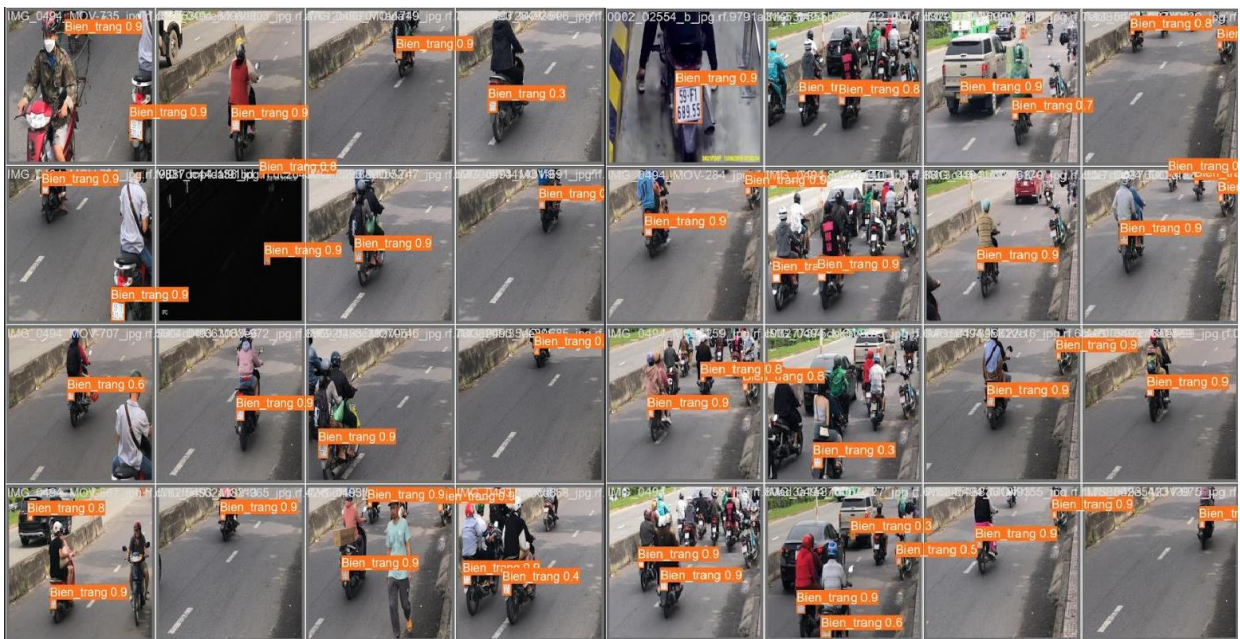
### 3.4 Đánh giá và nhận xét

#### 1. Bộ dữ liệu License-Plate

Mô hình cho kết quả dự đoán tốt với tập train cao với precision và recall trên 90%. Dựa vào Confusion Matrix các biển số bị dự đoán nhầm thành các background hoặc biển khác xảy ra rất ít, trong đó biển ngoại giao dự đoán tốt nhất với precision và recall trên 98%.

Tuy nhiên cũng có trường hợp background bị dự đoán thành biển trắng, có thể do trong quá trình thu thập dữ liệu không tốt, hoặc chưa bổ sung đầy đủ. Trên tập test, với IOU 0.5 mô hình cho kết quả đánh giá cao, với precision 96.2% , recall 95.4% và mAP50 97.8%. Cũng giống như trong tập train, mô hình cũng dự đoán tốt các biển số.

Các hình ảnh trong tập test sau khi dự đoán:



Hình 15: Các hình ảnh trong tập test sau khi dự đoán

#### 2. Bộ dữ liệu Character-Plate

Sau khi train mô hình cho kết quả nhận diện tốt với 32 ký tự chữ và số. Với các thông số như precision và recall tốt, nhìn chung đều trên 90%. Nhìn chung mô hình dự đoán tốt các ký tự trong đó có các ký tự số dự đoán tốt hơn so với các ký tự chữ. Ví dụ như từ 0 đến 9 cho độ chính xác trên 94%, còn ký tự chữ 'U', 'J', 'M', 'Y' cho độ chính xác trung bình. Lý do không đủ dữ liệu hoặc bị dự đoán nhầm thành các ký tự khác.

Đánh giá mô hình trên tập test với IOU là 0.5 cho độ chính xác cao. Cũng như trong tập train thì mô hình vẫn dự đoán tốt các ký tự số.

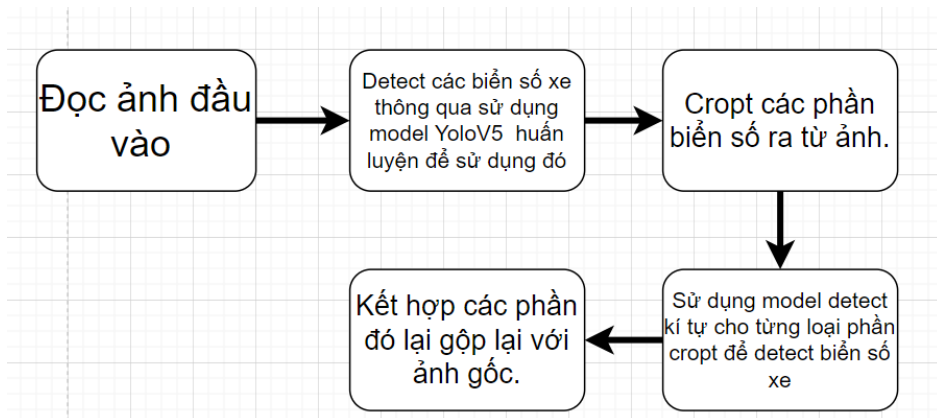




Hình 16: Các hình ảnh trong tập test sau khi dự đoán

### 3.5 Kết hợp 2 mô hình

Sơ đồ khối thuật toán:



Hình 17: Sơ đồ khối thuật toán

#### a. Detect các biển số xe thông qua sử dụng model YoloV5 huấn luyện sẵn.

-Để detect biển số xe, sử dụng mô hình YOLOv5 đã được huấn luyện. Tải mô hình từ trang chính thức và chuẩn bị ảnh hoặc video chứa xe. Chạy mô hình thông qua command line với các tham số như kích thước ảnh và ngưỡng tin cậy. Kiểm tra kết quả hiển thị đối tượng và độ tin cậy. Để đạt độ chính xác cao, cần có dữ liệu huấn luyện đủ chất lượng và điều chỉnh siêu tham số. Quá trình này giúp tự động detect biển số xe hiệu quả trong ảnh hoặc video. Sau đây là thành quả các việc detect:

	xmin	ymin	xmax	ymax	confidence	class	name
0	542.66803	402.814941	974.70697	888.899048	0.94192	1	Bien_trang

Hình 18: Ảnh thành quả sau khi detect được thông tin về biển số

**b. Crop biển số ra từ ảnh**

Với các thông tin đã được lấy ra từ model đã train trên đã lấy ra được thông tin về vị trí của biển số. Cắt ra thì được các ảnh như sau:



Hình 19: Tập hợp các biển số được detect

Nhận thấy là biển số xe có thể bị nghiêng nhưng có thể có rất nhiều kiểu nghiêng và góc nghiêng khác nhau nên cần phải làm thẳng lại thì mới đưa ra kết quả tốt nhất cho mô hình sau detect kí tự.

Xử lí xoay căn chỉnh lại biển số:

Các bước như sau:

Tìm hình chữ nhật bao quanh biển số bằng cách chuyển đổi hình ảnh thành hình ảnh xám, sử dụng ngưỡng nhị phân để phân đoạn ảnh và tìm các đường viền (contours) bằng phương pháp tìm đường viền. Sau đó, tìm hình chữ nhật bao quanh biển số có diện tích lớn nhất trong các đường viền.



Hình 20: ảnh trước khi được xoay chỉnh

Xác định góc xoay của hình chữ nhật bao quanh biển số bằng cách sử dụng phương pháp `minAreaRect()` để tìm hình chữ nhật nhỏ nhất bao phủ đường viền lớn nhất đã tìm được ở bước 1, và sau đó lấy góc nghiêng của hình chữ nhật này.





Hình 21: Đổi màu sau đó tìm ngưỡng của ảnh  
Xác định ma trận xoay để xoay hình ảnh ban đầu theo góc nghiêng tìm được ở bước 2 và thực hiện việc xoay hình ảnh ban đầu. Sau đó, tìm các đường viền trên hình ảnh đã xoay.



Hình 22 Xoay chỉnh ảnh

Tìm hình chữ nhật có diện tích lớn nhất trong các đường viền trên hình ảnh đã xoay và cắt bớt phần thừa để lấy hình ảnh của biển số. Sau khi thực hiện bước này sẽ thu được kết quả:



Hình 23 Ảnh sau cùng

**c. Sử dụng model detect kí tự cho từng phần crop để detect biển số xe**  
Sử dụng những ảnh đã detect trên để lấy ra những ảnh ra kí tự biển số:



	xmin	ymin	xmax	ymax	confidence	class name		
0	10.780644	6.879104	19.384405	26.325859	0.924372	0	0	
1	1.297350	5.013925	10.430493	24.181246	0.914065	8	8	
2	77.156754	20.230492	85.952881	41.379391	0.905104	9	9	
3	53.703423	14.040390	63.255455	37.254536	0.900011	7	7	
4	65.940369	15.914657	76.158218	39.578476	0.899516	7	7	
5	34.693748	11.911584	43.273983	31.408785	0.898660	0	0	
6	44.015732	13.139959	53.123352	35.214127	0.891451	2	2	
7	19.548018	8.160727	28.593119	31.618605	0.752955	10	A	
8	19.715813	7.924369	28.501534	31.297096	0.301447	4	4	

Hình 24: Kí tự detect ra từ ảnh được các số sau.

-Khi áp dụng vào những ảnh có độ phân giải thấp thì xảy ra trường hợp một kí tự lại được detect 2 lần. Đây là trường hợp overlap do chưa chọn lại giá trị ngưỡng. Cần xác định giá trị ngưỡng trung bình của toàn bộ tấm ảnh sau đó áp dụng vào trong các kí tự để lại bỏ những kí tự overlap có độ confidence thấp.

#### d. Kết hợp lại vào ảnh gốc

-Kết hợp với nhau và gán lên ảnh gốc sẽ ra được thành quả:



Hình 25: xe biển xanh nằm ngang



Hình 26: xe có biển số nằm nghiêng



Hình 27: Xe biển mờ

#### 4. TỔNG KẾT

Ưu điểm:

- Tính tự động cao: YOLOv5 cho phép detect cả biển số và kí tự biển số một cách tự động, giảm sự can thiệp của con người.
- Hiệu suất cao: Sử dụng mô hình đã được huấn luyện sẵn giúp tăng cường hiệu suất, giảm thời gian triển khai và đảm bảo độ chính xác.
- Linh hoạt: Có thể điều chỉnh các tham số như kích thước ảnh, ngưỡng tin cậy để tối ưu hóa cho môi trường sử dụng cụ thể.

Nhược điểm:

- Yêu cầu dữ liệu huấn luyện lớn: Để đạt kết quả tốt, cần có lượng dữ liệu huấn luyện đủ lớn và đa dạng về biển số và kí tự biển số.
- Chế độ ánh sáng và góc nhìn: Có thể gặp khó khăn khi áp dụng mô hình trong điều kiện ánh sáng và góc nhìn biến động.
- Điều chỉnh tham số: Yêu cầu sự điều chỉnh kỹ thuật các tham số như ngưỡng tin cậy để đảm bảo độ chính xác và tránh false positives hoặc false negatives.



## Tài liệu tham khảo

- [1]. Nhận diện kí tự trên biển số xe Việt Nam (<https://viblo.asia/p/nhan-dien-ki-tu-tren-bien-so-xe-viet-nam-2oKLnXWZ4QO>)
- [2]. Github YoloV5 (<https://github.com/ultralytics/yolov5>)
- [3]. Character-Time-series-Matching-For-Robust-License-Plate-Recognition (<https://github.com/chequanghuy/Character-Time-series-Matching?tab=readme-ov-file>)