# TÓM TẮT

Sự phát triển nhanh chóng của phương tiện giao thông ô tô đã gây ra một hạ tầng giao thông ngày càng phức tạp. Do đó, không chỉ loại biển báo mà cả số lượng biển báo trên đường cũng đang tăng lên. Dần dần, nhu cầu cho các ứng dụng hỗ trợ lái xe nhận biết biển báo trên đường để giúp họ tránh bỏ lỡ các biển báo cũng tăng lên.

Bài báo này đề xuất một phương pháp để phát hiện và phân loại các biển báo giao thông Việt Nam dựa trên thuật toán YOLO, một kiến trúc học sâu thống nhất cho các ứng dụng nhận diện thời gian thực. Tác giả đã thêm một thành phần tính toán kích thước anchor box dựa trên thuật toán phân cụm k-means để xác định kích thước anchor box cho thuật toán YOLO. Tác giả đã thu thập một bộ dữ liệu biển báo giao thông Việt Nam gồm 5000 hình ảnh chứa tổng cộng 5704 biển báo giao thông của nhiều loại để đánh giá.

Mô hình nhỏ, mặc dù là mô hình nhanh nhất nhưng cũng là mô hình ít chính xác nhất, đã đạt được F1 score trên 92% với thời gian phát hiện khoảng 0.17 giây (trong môi trường thử nghiệm của chúng tôi: laptop CPU Intel 3520M, 8GB RAM, không có GPU). So với các nghiên cứu tương tự về nhận diện biển báo giao thông Việt Nam, phương pháp đề xuất trong bài báo này thể hiện kết quả hứa hẹn, cung cấp một sự cân bằng tốt giữa độ chính xác và thời gian nhận diện cũng như khả năng thực tế cho các ứng dụng thực tế.

# MỤC LỤC

[TÓM TẮT iii](#_Toc160965381)

[DANH MỤC HÌNH VẼ vi](#_Toc160965382)

[DANH MỤC BẢNG BIỂU vii](#_Toc160965383)

[DANH MỤC CÁC CHỮ VIẾT TẮT viii](#_Toc160965384)

[CHƯƠNG 1. INTRODUCTION 9](#_Toc160965385)

[1.1 Lý do của bài báo: 9](#_Toc160965386)

[1.2 Mục tiêu thực hiện đề tài 9](#_Toc160965387)

[CHƯƠNG 2. PHÂN TÍCH LÝ THUYẾT VÀ MÔ HÌNH 10](#_Toc160965388)

[2.1 Bô dữ liệu (Vietnamese traffic signs): 10](#_Toc160965389)

[2.2 Giải thuật: 11](#_Toc160965390)

[2.3 Architecture: 11](#_Toc160965391)

[2.4 Identify Anchor Boxes Size: 14](#_Toc160965392)

[CHƯƠNG 3. Phân tích số liệu và đánh giá 16](#_Toc160965393)

[3.1 Bộ dữ liệu dùng đánh giá 16](#_Toc160965394)

[3.2 Kết quả đánh giá 17](#_Toc160965395)

[3.3 Đánh giá chung 20](#_Toc160965396)

[3.4 Hướng phát triển của mô hình 21](#_Toc160965397)

[CHƯƠNG 4. TỔNG QUAN LẠI VẤN ĐỀ VÀ DEMO HƯỚNG ỨNG DỤNG LIÊN QUAN ……………………………………………………………………...21](#_Toc160965398)

[4.1 Tổng quan: 21](#_Toc160965399)

[4.2 Các phương pháp khả dụng khác: 22](#_Toc160965400)

[4.3 Nhận định và lựa chọn phương pháp giải quyết: 23](#_Toc160965401)

[4.4 Khái niệm, thuật ngữ cơ bản áp dụng: 24](#_Toc160965402)

[4.5 Tổng hợp lại về YOLO: 25](#_Toc160965403)

[4.6 Mô tả demo 27](#_Toc160965404)

[4.7 Kết quả thực nghiệm 35](#_Toc160965405)

**TÀI LIỆU THAM KHẢO**……………………………..………………………....37

# DANH MỤC HÌNH VẼ

**Hình 1**. Example of Vietnamese traffic signs ………………………………………9

**Hình 2**. Traffic sign in a dashcam image ………………………………………….10

**Hình 3**. Architecture of the Vietnamese tra c sign recognition system …………...11

**Hình 4**. Công thức IoU ……………………………………………………………12

**Hình 5**. The YOLO architecture. ………………………………………………….13

**Hình 6**. The bounding box sizes clustering by k-means algorithm ……………….13

**Hình 7**. Calculate the final bounding boxes ………………………………………14

**Hình 8**. Biển báo giao thông dùng thử nghiệm …………………………………...22

**Hình 9**. Mô Hình Yolo ……………………………………………………………24

**Hình 10**. Hình ảnh hoạt động của mô hình …………………………….…...…….25

**Hình 11**. Hàm tính IOU …………………………………………………………..26

**Hình 12**. Cách cài thư viện cần thiết ……………………………………………...27

**Hình 13**. Cách kết nối tập dữ liệu ………………………………………………...28

**Hình 14**. Kết nối với model của mô hình ……………………….………………...28

**Hình 15**. Tạo thư mục train ……………………………………………….………28

**Hình 16**. Phân chia lớp biển báo ……………………………………………….…29

**Hình 17**. Cách chuyển file bằng pandas …………………………………………..29

**Hình 18**. Cấu trúc lại file CSV…………………………………………………….30

**Hình 19.** Chuyển đổi đuôi file……………………………………………………..31

**Hình 20**. Kiểm tra và loại bỏ đuôi file……………………………………………..32

**Hình 21**. Tạo file config……………………………………………………………32

**Hình 22**. Gọi model và upload file thiết lập……………………………………….32

**Hình 23**. Thông tin lớp và thông số………………………………………………..33

**Hình 24**. Thông tin thông số của mỗi vòng chạy…………………………………..34

**Hình 25**. Thông tin về model , tốc độ và file chứa…………………………………34

**Hình 26**. Cách thiết lập lại locade………………………………………………….34

**Hình 27**. Cách trực quan hóa kết quả thực nghiệm…………………………………35

# DANH MỤC BẢNG BIỂU

**Bảng 1**: Bảng thống kê dữ liệu dùng để huấn luyện và thử nghiệm……………….16

**Bảng 2**: Bảng thống kê dữ liệu sau thử nghiệm…………………………………...18

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **CNN** | Convolutional neural networks |
| **CS** | Computer Vision |

# INTRODUCTION

## Lý do của bài báo:

Sự phát triển của phương tiện ô tô đang tác động đến cơ sở hạ tầng giao thông một cách toàn diện. Sự gia tăng về số lượng và loại hình biển báo đường phản ánh xu hướng phát triển mới này. Những người lái xe ngày nay đối mặt với việc phải nhớ nhiều biển báo hơn và phải tập trung hơn vào việc quan sát biển báo trong quá trình lái xe, điều này rất quan trọng để đảm bảo an toàn giao thông. Vì vậy, sự xuất hiện của hệ thống phát hiện và phân loại biển báo giao thông thời gian thực trở nên cấp bách, nhằm hỗ trợ tài xế và giúp họ tránh xa các tình huống nguy hiểm. Ngoài ra, công nghệ nhận diện biển báo cũng có thể được ứng dụng rộng rãi trong việc phát triển các loại xe thông minh hoặc xe tự lái, mở ra nhiều tiềm năng ứng dụng trong tương lai.

## Mục tiêu thực hiện đề tài

Kể từ những năm 1990, nhiều nghiên cứu đã tập trung vào việc nhận diện biển báo giao thông. Các phương pháp thường sử dụng kỹ thuật xử lý ảnh hoặc các thuật toán học máy để nhận diện và phân loại biển báo [3, 11, 6, 2, 9, 5].

Trong những năm gần đây, với sự phát triển của học sâu, nhiều nghiên cứu mới đã áp dụng học sâu để cải thiện hiệu suất nhận diện biển báo [4, 7, 12, 17, 15]. Các mô hình mạng nơ-ron sâu đã đạt được những kết quả ấn tượng trong lĩnh vực này.

Bài báo này giới thiệu một mô hình nhận diện biển báo giao thông Việt Nam cùng với kết quả đánh giá hiệu suất của nó, cung cấp một cái nhìn tổng quan về những tiến triển trong lĩnh vực nhận diện biển báo giao thông và định hướng cho những nỗ lực nghiên cứu trong tương lai.

# PHÂN TÍCH LÝ THUYẾT VÀ MÔ HÌNH

## Bô dữ liệu (Vietnamese traffic signs):

Xét tại ngữ cảnh và thời gian ra đời của bài báo thì hệ thống biển báo giao thông Việt Nam có khoảng 144 biển báo giao thông khác nhau. Chúng nó được chia thành 4 nhóm lớn: prohibitory or restrictive signs (40 biển báo), warning signs (47 biển báo), mandatory signs (10 biển báo) và indication signs (47 biển báo). Hình mẫu dưới đây mô tả một vài biển báo giao thông của 4 loại

A screenshot of a road sign

Description automatically generated

**Hình 1.** Example of Vietnamese traffic signs

A street with a sign on it

Description automatically generated**Hình 2.** Traffic sign in a dashcam image

## Giải thuật:

Bài toán Recognition Traffic Signs bao gồm 2 vấn đề chính cần giải quyết:

* Detect the traffit signs within the images
* Classify the detected signs

Trong bài báo này chúng ta sẽ xay dựng mô hình để nhận diện biển báo giao thông dựa trên giải thuật YOLO. Đồng thời chúng ta cũng sử dụng giải thuật kmean để phân loại các bounding box để xác định được kích thước của anchor.

Dữ liệu huấn luyện sẽ phải gồm dữ liệu nhãn là các bounding boxes và nhãn của biến báo giao thông ứng với từng ảnh trong dữ liệu ảnh. Chúng ta sử dụng giải thuật kmean nhận dữ liệu nhan để tính toán ra anchor size. Anchor size được tính từ kmean cùng với dư liệu nhãn sẽ được đưa vào YOLO để xây dựng mô hình nhận diện biển báo giao thông.

## Architecture:

YOLO được công bố năm 2015 bởi Joseph Redmon và bạn của ông. Nó vô cùng phù hợp cho việc phát hiện đối tượng theo thời gian thực chỉ thông qua việc mạng nơ ron đơn nhất sẽ dự đoán xác suất của bounding boxes và lớp trong cả bức ảnh.

A diagram of a machine

Description automatically generated

**Hình 3.** Architecture of the Vietnamese tra c sign recognition system

Tất cả các đặc chưng trong hình đều sẽ tham gia vào quá trình dự doán các bounding box. Tất cả đều bounding box sẽ được dự đoán xác suất thuộc về từng class. Tiếp theo sẽ là các bước thực hiện của YOLO:

* Chia ảnh thành các grid cells (các ô vuông nhỏ đều nhau ) thường là 1 ma trận vuông dạng S x S grids.
* Mỗi cell sẽ được dự đoán B bounding boxes cùng với confidence score của nó.
* Mỗi bounding box sẽ gồm những thông tin được dự đoán như (bx,by,bw,bh,p)
* Trong đó (bx,by) là tọa độ của tâm của bounding box so với cell
* Còn (bw,bh) là chiều cao và chiều rộng của bounding box so với toàn ảnh
* Cuối cùng là dự đoán confidence score p với công thức:

p = P r(Object) × IOUtruthpred

A diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of

Description automatically generatedPr(Object) dao động từ 0->1 và bằng 1 khi phần diện tích của đối tượng đúng nằm trong nó và 0 nếu ngược lại. IOUtruthpred là intersection over union giữa pred box và truth box:

**Hình 4.** Công thức IoU

Mỗi một grid cell sẽ có các xác suất conditional class riêng C= Pr(Class i | Object), đây là giá trị thể hiên được xác suất để biết các objects trong grid cell có cái nào thuộc class đang xét không.

Thời điểm nhận diện, C x p = Pr(Class i | Object) x P r(Object) × IOUtruthpred cho chúng ta biết được khả năng thuộc về class nào của từng box.Chỉ số trên không chỉ thể hiện được xác suất class xuất hiện trong box mà còn cho chúng ta biết độ tốt của box dự đoán so với vật thể.

Mô hình trên được xây dựng dựa trên CNN. Các lớp convolution sẽ trích xuất đặc trưng từ ảnh trong khi các lớp fully connected sẽ dự đoán tọa độ và xác suất.

Có rất nhiều phiên bản khác nhau của YOLO nhưng chúng thường chỉ khác nhau về số lượng convolution layer và fully connected layer.

Trong ngữ cảnh của bài báo thì do điều kiện về phần cứng hạn chế nên sẽ chỉ sử dụng phiên bản thấp của thuật toán YOLO là YOLOv3 hay gọi là “tiny YOLO v3”.

Phiên bản YOLOv3 có 13 lớp tích chập cùng với nhiều lớp phụ hỗ trợ để trích xuất đặc trưng của ảnh.YOLOv3 vượt trội hơn vì khả năng nhận diện dc các đối tượng nhỏ hơn.

A diagram of a diagram of different sizes of boxes

Description automatically generated

**Hình 5.** The YOLO architecture.

## Identify Anchor Boxes Size:

Để có được anchor boxes size trong ngữ cảnh bài thì sẽ sử dụng thuật toán k-mean để xác định. Với YOLOv3 chúng ta sẽ nhận diện boxes qua 2 tỷ lệ scale khác nhau (theo em hiểu là bước chia grid cells) thành 2 tỷ lệ là: 13x13 và 26x26. Với mỗi tỷ lệ YOLOv3 sẽ sử dụng 3 loại anchor boxes.Do có 2 tỷ lệ mỗi tỷ lệ 3 loại nên cần 2x3=6 kích thước anchor box khác nhau cho YOLOv3.Để tính được giá trị cần thiết thì kích thước của tất cả bounding boxes trong dữ liệu train phải được phân thành 6 cụm khác nhau cụ thể trong bài là thông qua thuật toán k-means.

A colorful dots in different colors

Description automatically generated with medium confidence

**Hình 6.** The bounding box sizes clustering by k-means algorithm

Trong bài, k-means phân bộ dữ liệu train thành 6 cụm anchor boxes sizes khác nhau lần lượt là (20,39), (51,28) , (37,72), (89,48), (116,61) and (180,95).

Các kích thước anchor boxes cũng được sử dụng trong bước detection.Trong lớp detection nằm trong mỗi tỷ lệ scale thì kích thước của bounding boxes (bx,by,bw,bh) sẽ được tính toán dựa trên các giá trị trong feature map (tx,ty,tw,th). Chiều cao và rộng sẽ được tính toán bằng hàm tuyến tính:

bw = pw + e tw

bh = ph + e th

pw và ph là chiều cao và rộng của anchor box tương ứng. Tâm của đối tượng (bx,by) là tổng của sigmoid(tx,ty) và offset (cx,cy) .

(cx,cy) là tọa độ của grid cell chứa đối tượng. Ví dụ, tâm của đối tượng nằm trong grid cell mà có sự giao nhau giữa dòng thứ 1 và cột thứ 2 thì chúng ta có cx=1,cy=2.

bx = σ(tx) + cx

A blue square with black lines and letters

Description automatically generatedby = σ(ty) + cy

**Hình 7.** Calculate the final bounding boxes

# Phân tích số liệu và đánh giá

## Bộ dữ liệu dùng đánh giá

- Tập dữ liệu dùng huấn luyện mô hình được lấy từ thành phố Cần Thơ trong quá trình di chuyển bằng camera điện thoại

- Tập dữ liệu bao gồm 5704 hình ảnh của 33 loại biển báo giao thông. Các biển báo tương tự được nhóm lại với nhau để giảm số lớp học và được gắn nhãn hình ảnh

- 80℅ (gồm 4539 ảnh)  là số lượng ảnh dùng để huấn luyện cho mô hình Yolo

- 20% (gồm 1165 ảnh ) là số lượng ảnh còn lại dùng để kiểm thử mô hình

Tập dữ liệu bao gồm 5704 hình ảnh đã được thu thập và 22 lớp được mô tả sau đây:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Label** | **Description** | **Train** | **Test** |
| 1 | 102 | No entry | 204 | 22 |
| 2 | 130 | No stopping or parking or waiting | 268 | 66 |
| 3 | 131 | No parking or waiting | 180 | 54 |
| 4 | 201a | Curve to the left | 100 | 36 |
| 5 | 201b | Curve to the right | 130 | 45 |
| 6 | 202 | Series of curves | 180 | 41 |
| 7 | 203 | Road narrows | 54 | 35 |
| 8 | 205 | Road junction | 307 | 79 |
| 9 | 207 | Road junction with priority | 773 | 182 |
| 10 | 208 | Road junction with priority | 145 | 29 |
| 11 | 209 | Traffic signals ahead | 125 | 25 |
| 12 | 221 | Rough road surface | 110 | 16 |
| 13 | 224 | Pedestrian crossing ahead | 428 | 100 |
| 14 | 225 | Children | 425 | 116 |
| 15 | 233 | Other danger | 111 | 24 |
| 16 | 245 | Slow | 62 | 12 |
| 17 | 302 | Keep right | 136 | 54 |
| 18 | 303 | Roundabout | 108 | 23 |
| 19 | 423 | Pedestrian crossing | 310 | 90 |
| 20 | crowded | Start of a crowded area | 50 | 11 |
| 21 | end\_crowded | End of a crowded area | 29 | 10 |
| 22 | Tracffic\_light | Traffic light | 304 | 95 |
| **Total** | | | **4539** | **1165** |

**Bảng 1:** Bảng thống kê dữ liệu dùng để huấn luyện và thử nghiệm

Ở bảng 1 thấy, ta có thể nhận thấy loại biển báo "Road junction with priority" thuộc label 207 là loại dữ liệu có số lượng nhiều nhất (773 ảnh dùng huấn luyện, 182 ảnh dùng thử nghiệm),

- Một số nhãn như “Start of a crowded area” hoặc "End of a crowded area" (29 ảnh dùng huấn luyện, 10 ảnh dùng thử nghiệm) có số lượng mẫu rất ít, có thể gây khó khan cho mô hình nhận diện chính xác các biển báo hiếm này.

-Dữ liệu đào tạo và kiểm thử cho mỗi nhãn có sự chênh lệch về số lượng. Điều này có thể ảnh hưởng đến hiệu suất của mô hình, đặc biệt là nếu có một số lớn mẫu trong đào tạo và ít mẫu trong kiểm thử.

Vậy để đảm bảo hiệu suất tốt nhất của mô hình, quan trọng là cân nhắc điều chỉnh kích thước và độ đa dạng của dữ liệu đào tạo, đồng thời kiểm tra và đánh giá mô hình trên tập kiểm thử đại diện

## Kết quả đánh giá

Mô hình phát hiện tín hiệu giao thông đã được sử dụng các cách đo đánh giá sau :

**+ True Positive (TP):** mô hình nhận dạng loại biển báochính xác, tức là mô hình dự đoán chính xác tên lớp và giá trị IoU giữa predicted bounding box và ground truth box phải lớn hơn hoặc bằng 0,5. (IoU thông thường được dùng là 0.5 nhưng có thể được thay đổi tùy thuộc vào yêu cầu cụ thể của bài toán và đặc tính của dữ liệu, ở trong mô hình này nếu IoU thấp hơn 0.5 thì dự đoán được tính là sai).

**+ False Negative (FN):** biển báo giao thông tồn tại trong ảnh, nhưng mô hình không thể nhận dạng nó

Cụ thể, FN xuất hiện khi:

* Biển báo không được dự đoán: Mô hình không tạo ra bounding box hoặc dự đoán rằng không có biển báo trong ảnh.
* Dự đoán không chính xác: Nếu có bounding box được tạo ra, nhưng nó không chính xác về đối tượng thực tế.

(False Negative thường là một vấn đề quan trọng trong bài toán nhận diện biển báo giao thông vì nó có thể dẫn đến sự bỏ lỡ các tình huống quan trọng và gây nguy hiểm trong giao thông. Để giảm thiểu FN, cần tối ưu mô hình và tăng cường dữ liệu đào tạo để nâng cao khả năng nhận diện đối tượng)

**+ False Positive (TP**): là tình huống mà mô hình nhận diện biển báo sai , dự đoán các đối tượng không phải biển báo ( vật thể, cảnh đẹp hoặc nhiễu) hoặc lớp đối tượng đúng nhưng giá trị Iou giữa predicted bounding box và ground truth box ít hơn 0.5

Như đã nói , thì mô hình sử dụng 80% dữ liệu thu thập được để huấn luyện và 20% dữ liệu còn lại được dùng để kiểm thử. Bảng sau là kết quả đánh giá khi sử dụng 20% dữ liệu còn lại

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Traffic sign | True Positive | False Negative | False Positive |
| 1 | 102 | 22 | 0 | 0 |
| 2 | 130 | 61 | 4 | 1 |
| 3 | 131 | 54 | 0 | 0 |
| 4 | 201a | 21 | 0 | 15 |
| 5 | 201b | 36 | 0 | 9 |
| 6 | 202 | 41 | 0 | 0 |
| 7 | 203 | 12 | 1 | 22 |
| 8 | 205 | 78 | 0 | 1 |
| 9 | 207 | 180 | 1 | 1 |
| 10 | 208 | 29 | 0 | 0 |
| 11 | 209 | 25 | 0 | 0 |
| 12 | 221 | 16 | 0 | 0 |
| 13 | 224 | 99 | 1 | 0 |
| 14 | 225 | 115 | 0 | 1 |
| 15 | 233 | 23 | 0 | 1 |
| 16 | 245 | 12 | 0 | 0 |
| 17 | 302 | 40 | 14 | 0 |
| 18 | 303 | 20 | 2 | 1 |
| 19 | 423 | 90 | 0 | 0 |
| 20 | Crowded | 9 | 0 | 2 |
| 21 | end\_crowded | 5 | 1 | 4 |
| 22 | traffic\_light | 1 | 94 | 0 |
| Total | | 989 | 118 | 58 |

**Bảng 2:** Bảng thống kê dữ liệu sau thử nghiệm

Về False Negative thì số lượng FN khá thấp, đặc biệt là với nhiều loại biển báo, có nghĩa là mô hình có khả năng nhận diện các biển báo này khá tốt.

Về False Positvie thì số lượng FP xuất hiện, chẳng hạn với các biển báo "Curve to the left" (201a), "Curve to the right" (201b), và "Road junction" (205). Việc này có thể đòi hỏi cải thiện để giảm số lượng FP và tăng độ chính xác của mô hình

Điểm cần cải thiện: Mô hình gặp khó khăn trong việc nhận diện biển báo "traffic\_light" với số lượng FN là 94 và TP chỉ có 1. Có thể cần kiểm tra và cải thiện khả năng nhận diện đối với loại biển báo này.

Tóm lại, mặc dù có một số điểm có thể cải thiện, nhưng tổng thể, mô hình có hiệu suất khá tốt trong việc nhận diện biển báo giao thông. Việc điều chỉnh và tối ưu hóa mô hình dựa trên các chỉ số đánh giá sẽ giúp cải thiện chất lượng dự đoán.

* **Đánh giá chung hiệu suất mô hình :**

precision và recall của mô hình khá cao khi pre là 0.95 còn recall là 0.89

==

Ta thấy trong số 1063 trường hợp biển báo  được dự đoán, chỉ có 58 trường hợp biển báo giao thông (khoảng 11% là xác định sai) được thể hiện qua precision). Việc kết quả xét nghiệm chưa đủ tin cậy (trong trường hợp này là phân lớp sai – “báo động nhầm”) khiến các phương tiện ứng dụng có thể sẽ phải sử phương pháp khác có độ chính xác cao hơn , gây lãng phí thời gian , đồng thời ảnh hưởng trực tiếp đến quá trình lái xe làm gây sai phạm trong quá trình giao thông

Với giá trị recall = 0.89, ta có thể hiểu rằng trong số 1112 trường hợp nhận dạng biển báo giao thông , mô hình  phát hiện ra 1005 trường hợp (khoảng 89%). Trong thực tế, những đối tượng bị bỏ sót bởi mô hình luôn có khả năng khiến cho người di chuyển mắc sai phạm giao thông . Có thể dễ dàng nhận ra rằng cần phải  nâng cấp cải thiện Recall

## Đánh giá chung

- Mô hình có thời gian nhận dạng rất nhanh cho từng ảnh (0.17 giây)

- Mô hình có sự cân bằng giữa độ chính xác và thời gian khi sử dụng Yolo V3 vì YOLO 3 nổi tiếng với khả năng xác định đối tượng nhanh chóng do việc dự đoán trực tiếp trên toàn bức ảnh. Điều này làm cho nó rất thích hợp cho các ứng dụng thời gian thực.

- Độ chính xác về nhận dạng của mô hình còn thấp so với những mô hình hiện đại

nguyên nhân là có thể do độ phân bố các dữ liệu thiếu cân bằng, không đồng đều

- Vì phần cứng của máy tính còn hạn chế, nên chỉ sử dụng phiên bản dùng tối ưu hóa thời gian phát hiện

## Hướng phát triển của mô hình

-Đối với các tài xế lái xe , ta có thể nâng cấp kết quả nhận diện thành dạng âm thanh (audio) để thuận tiện nó có thể trở thành trợ lý cho các tài xế lái xe, củng cố hơn về độ an toàn cho người dùng giao thông thông và giảm thiểu nguy cơ xảy ra tai nạn giao thông Ngoài ra ta có thể cung cấp thêm dữ liệu về các loại biển báo và cách phản ứng của thiết bị khi nhận diện chúng ( bổ sung them như là vạch kẻ đường, hay các hiệu lệnh của những người điều khiển giao thông ) để có thể ứng dụng cho khả năng tự động lái xe

-Vì đây là một đề tài có tính thiết thực cao, khi mà bối cảnh giao thông đang phức tạp hiện nay ở việt nam, ta có thể hướng đến các phiên bản mới nhất của YOLO để cải thiện những thiếu sót còn mắc phải cũng như là có thể áp dụng cho các biển báo trên toàn thế giới thì YOLOv8 cùng với bộ dữ liệu biển báo giao thông có sẵn của (GTSRB - German Traffic Sign Recognition Benchmark) sẽ đáp ứng các nhu cầu trên thông qua chương 4.

# TỔNG QUAN LẠI VẤN ĐỀ VÀ DEMO HƯỚNG ỨNG DỤNG LIÊN QUAN

## Tổng quan:

Phân tích vấn đề: Từ nội dung yêu cầu của đề tài ta rút chích ra được các vấn đề cần giải quyết sau:

* Cần xử lý dạng dữ liệu là biển báo giao thông nên dạng dữ liệu tiếp xúc là ảnh nên cần chuẩn bị dữ liệu về ảnh có chứa biển báo giao thông
* Đầu ra yêu cầu là ảnh đã được khoanh vùng các biển báo bằng khung hình chữ nhật một cách tự động nên đây cần phải xử lý dạng bài toán object detection hay cụ thể hơn là quy về bài toán traffic sign detection
* Ngoài ra với yêu cầu xuất nội dung biển báo ở đây chứng tỏ chúng ta cần thêm xử lý về mặt phân loại biển báo dạng bái toán classification

## Các phương pháp khả dụng khác:

***Về vấn đề dữ liệu:***

* Chúng ta có khá nhiều sự lựa chọn từ các bộ dataset có sẵn trên các trang diễn đàn về công nghệ thông tin (ex: kaggle, github...).
* Tự thiết kế dữ liệu bằng các hình ảnh trên mạng hoặc tự kiếm.

***Về vấn đề object detection:***

* Chúng ta có nhiều loại mô hình khác nhau từ học máy đến học sâu nhưng theo khảo sát thì phù hợp và tốt nhất dành cho đối tượng của đề tài em lọc ra được các kiến trúc sau:
* R-CNN
* YOLO
* SOLO
* HAAR CASSCADE
* .....

***Về vấn đề phân loại:***

* Chúng ta có nhiều loại hình thức khác nhau và phương pháp khác nhau để phân loại ảnh từ các thuật toán cơ sở học máy đến các kiến trúc bậc cao của cnn để xử lý đối tượng là ảnh.
* Ngoài ra tối ưu hơn nữa với một vài kiến trúc học sâu thì việc classification được tích hợp sẵn thông qua một thuật toán phổ biến mà ai cũng biết là CNN.

***Về vấn đề phát sinh về xử lý:***

* Chúng ta có thể thực hiện xử lý trực tiếp trên máy tình bằng nhưng sẽ có các vấn đề dễ phát sinh như độ phức tạp thuật toán cao dẫn đến khó khăn trong việc thực thi, xử lý với bộ dữ liệu ảnh khá nhiều dễ bị tràn hay không đủ dung lượng ảnh hưởng đến hiệu quả của bài làm.
* Chúng ta có thể sử dụng máy ảo trung gian để thực thi, ví dụ như nổi bật nhất phổ biến nhất là (google colab) và lưu trữ dữ liệu qua google drive để tiện lợi cho việc sử dụng.

## Nhận định và lựa chọn phương pháp giải quyết:

***Về vấn đề môi trường thực thi:***

Em sẽ thực hiện qua google colab và dữ liệu được đẩy lên google drive.

***Về vấn đề object detection:***

Trong bài này phương pháp em chọn sẽ là YOLO một mô hình được giá tốt nhất cũng như mới hơn so với mặt bằng chung nhằm tối ưu hóa đầu ra tốt nhất.

***Về vấn đề classification:***

Trong bài này phương pháp em chọn sẽ là thuật toán CNN kết hợp trong kiến trúc YOLO để giúp cho mô hình được đồng bộ và mang lại hiệu quả khả quan hơn khi gán nhãn cho đối tượng.

***Về vấn đề dữ liệu:***

Trong bài demo báo cáo em sẽ sử dụng bộ dữ liệu GTSDB thay thế cho bộ dữ liệu Vietnamese traffic signs để có góc nhìn khái quát hơn về mô hình đối với dữ liệu khác :

<https://sid.erda.dk/public/archives/ff17dc924eba88d5d01a807357d6614c/TrainIJCNN2013.zip>

A group of cars parked on a street

Description automatically generatedA collection of road signs

Description automatically generatedA road with signs on it

Description automatically generatedVà dữ liệu để test thử nghiệm em sẽ chon 3 ảnh sau để thực hiện:

**Hình 8.** Biển báo giao thông dùng thử nghiệm

## Khái niệm, thuật ngữ cơ bản áp dụng:

Dữ liệu GTSBD:

* Gồm 600 bức ảnh dạng .ppm về cảnh đường phố giao thông tự nhiên của Đức théo thứ tự 00000.ppm -> 00599.ppm.
* Gồm 1 file gt.txt chứa labels và vị trí bounding box của từng ảnh.
* File gt.txt sẽ chứa nhãn của từng bounding box đúng của ảnh và phân thành 43 loại khác nhau:

0 = speed limit 20 (prohibitory)

1 = speed limit 30 (prohibitory)

2 = speed limit 50 (prohibitory)

3 = speed limit 60 (prohibitory)

4 = speed limit 70 (prohibitory)

5 = speed limit 80 (prohibitory)

6 = restriction ends 80 (other)

7 = speed limit 100 (prohibitory)

8 = speed limit 120 (prohibitory)

9 = no overtaking (prohibitory)

10 = no overtaking (trucks) (prohibitory)

11 = priority at next intersection (danger)

12 = priority road (other)

13 = give way (other)

14 = stop (other)

15 = no traffic both ways (prohibitory)

16 = no trucks (prohibitory)

17 = no entry (other)

18 = danger (danger)

19 = bend left (danger)

20 = bend right (danger)

21 = bend (danger)

22 = uneven road (danger)

23 = slippery road (danger)

24 = road narrows (danger)

25 = construction (danger)

(other)

26 = traffic signal (danger)

27 = pedestrian crossing (danger)

28 = school crossing (danger)

29 = cycles crossing (danger)

30 = snow (danger)

31 = animals (danger)

32 = restriction ends (other)

33 = go right (mandatory)

34 = go left (mandatory)

35 = go straight (mandatory)

36 = go right or straight (mandatory)

37 = go left or straight (mandatory)

38 = keep right (mandatory)

39 = keep left (mandatory)

40 = roundabout (mandatory)

41 = restriction ends (overtaking)

42 = restriction ends

(overtaking (trucks)) (other)

Trong đó 43 class sẽ được phân bố vào 3 class lớn hơn là:

# Circular, white ground with red border:

prohibitory = [0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 15, 16]

# Circular, blue ground:

mandatory = [33, 34, 35, 36, 37, 38, 39, 40]

# Triangular, white ground with red border:

danger = [11, 18, 19, 20 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]

# And the other:

Other= [ left over ]

## Tổng hợp lại về YOLO:

* ***Khái niệm:***

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp 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.

A diagram of a number of boxes

Description automatically generated

**Hình 9.** Mô Hình Yolo

* ***Cách hoạt động:***
* Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành S x S ô thường là 3x3,7x7,9x9,..

A diagram of a dog and a bicycle

Description automatically generated

**Hình 10.** Hình ảnh hoạt động của mô hình

* Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành S x S ô thường là 3x3,7x7,9x9,..
* Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước S × S × ( 5 × N + M ) S×S×(5×N+M) với số lượng tham số mỗi ô là ( 5 × N + M ) (5×N+M) với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Ví dụ với hình ảnh trên chia thành 7 × 7 7×7 ô, mỗi ô cần dự đóan 2 bounding box và 3 object : con chó, ô tô, xe đạp thì output là 7 × 7 × 13 7×7×13, mỗi ô sẽ có 13 tham số, kết quả trả về ( 7 × 7 × 2 = 98 ) (7×7×2=98) bounding box.
* Dự đoán mỗi bounding box gồm 5 thành phần (x, y, w, h, prediction) với (x, y ) là tọa độ tâm của bounding box, (w, h) lần lượt là chiều rộng và chiều cao của bounding box, prediction được định nghĩa Pr(Object)∗ IOU(pred,truth). Với hình ảnh trên như ta tính mỗi ô sẽ có 13 tham số, ta có thể hiểu đơn giản như sau tham số thứ 1 sẽ chỉ ra ô đó có chứa đối tượng nào hay không P(Object), tham số 2, 3, 4, 5 sẽ trả về x, y, w, h của Box1. Tham số 6, 7, 8, 9, 10 tương tự sẽ Box2, tham số 11, 12, 13 lần lượt là xác suất ô đó có chứa object1 (P(chó|object), object2 (P(ô tô|object)), object3(P( xe đạp|object)). Lưu ý rằng tâm của bounding box nằm ở ô nào thì ô đó sẽ chứa đối tượng, cho dù đối tượng có thể ở các ô khác thì cũng sẽ trả về là 0. Vì vậy việc mà 1 ô chứa 2 hay nhiều tâm của bouding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế củamô hình YOLO1, vậy ta cần phải tăng số lượng ô chia trong 1 ảnh lên đó là lí do vì sao mình nói việc chia ô có thể làm ảnh hưởng tới việc mô hình phát hiện đối tượng.

***Hàm tính IOU:***

A diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of

Description automatically generated

**Hình 11**. Hàm tính IOU

***The Network:***

A screenshot of a computer code

Description automatically generated

## Mô tả demo

* Cài đặt các thư viện cần thiết để thực hiện bài toán:

A screenshot of a computer

Description automatically generated

**Hình 12**. Cách cài thư viện cần thiết

* Bước kết nối đến tập dữ liệu đã chuẩn bị và upload lên google drive, extract file:

A close-up of a text

Description automatically generated

A close-up of a car

Description automatically generated

**Hình 13.** Cách kết nối tập dữ liệu

* Chúng ta lấy về model pre-trained yolov8 cụ thể hơn là yolov8x ngoài ra còn các loại mô hình khác như yolov8n nhưng kết quả tốt nhất là yolov8x:

**Hình 14.** Kết nối với mode của mô hình

* Tạo thư mục train và 2 thư mục con images và labels lần lượt chưa labels và dữ liệu ảnh sau đó phân bổ các file về đúng vị trí từ dữ liệu đã giải nén phía trên:

A screenshot of a computer

Description automatically generated

**Hình 15.** Tạo thư mục train

* Tiếp theo chúng ta phân 43 lớp biển báo nhỏ thành 4 lớp biển báo lớn:[Prohibitory, Danger, Mandatory, Other]

A screenshot of a computer screen

Description automatically generated**Hình 16.** Phân chia lớp biển báo

* Chuyển file gt.txt thành cấu trúc file csv bằng pandas:

A screenshot of a computer

Description automatically generated

**Hình 17.** Cách chuyển file bằng pandas

* Cấu trúc lại file csv, điều chỉnh id về lại 4 lớp chính từ 43 lớp, tính toán lại tâm của bounding box cho cả x và y và chiều cao, chiều rộng của bounding box với tất cả các dòng:

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

**Hình 18.** Cấu trúc lại file CSV

* Tìm và chuyển các file có đuôi ‘.ppm’ về thành file ‘.jpg’ và file csv mà ta đã cấu trúc trên thành các file ‘.txt’ nhỏ tương ứng với từng ảnh sau đó phân bổ nó về 2 file images và labels trong thư mục train:

A screen shot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Hình 19.** Chuyển đổi đuôi file

* Kiểm tra và loại bỏ hết các file ‘.ppm’

A screenshot of a computer code

Description automatically generated

**Hình 20.** Kiểm tra và loại bỏ đuôi file

* Tiếp theo chúng ta cần tạo file config cho mô hình với dạng file đuôi ‘.yaml’ cụ thể ở đây là file ‘mydataset.yaml’ với thiết lập đường dẫn đến bộ dữ liệu của mình để train đồng thời cũng cần set số lớp biển báo và tên của từng nhãn ở đây ( nc: 4 – số lớp biển báo là 4) (names: [.....] – nhãn tên của từng lớp):

A computer code with text

Description automatically generated

**Hình 21.** Tạo file config

* Tiếp theo chúng ta sẽ gọi model ‘ yolov8x.pt’ lên và bắt đầu đưa file thiết lập phía trên kèm theo nhứng thiết lập cần thiết vào và bắt đầu train mô hình ở đây em sẽ train mô hình qua 5 epochs:

A screenshot of a computer code

Description automatically generated

**Hình 22.** Gọi model và upload file thiết lập

* Setup mặc định của model đang train:

task=detect, mode=train, model=yolov8x.pt, data=final/mydataset.yaml, epochs=5, time=None, patience=50, batch=16, imgsz=640, save=True, save\_period=-1, cache=False, device=None, workers=8, project=None, name=train7, exist\_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single\_cls=False, rect=False, cos\_lr=False, close\_mosaic=10, resume=False, amp=True, fraction=1.0, profile=False, freeze=None, overlap\_mask=True, mask\_ratio=4, dropout=0.0, val=True, split=val, save\_json=False, save\_hybrid=False, conf=None, iou=0.7, max\_det=300, half=False, dnn=False, plots=True, source=None, vid\_stride=1, stream\_buffer=False, visualize=False, augment=False, agnostic\_nms=False, classes=None, retina\_masks=False, embed=None, show=False, save\_frames=False, save\_txt=False, save\_conf=False, save\_crop=False, show\_labels=True, show\_conf=True, show\_boxes=True, line\_width=None, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify=False, opset=None, workspace=4, nms=False, lr0=0.01, lrf=0.01, momentum=0.937, weight\_decay=0.0005, warmup\_epochs=3.0, warmup\_momentum=0.8, warmup\_bias\_lr=0.1, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label\_smoothing=0.0, nbs=64, hsv\_h=0.015, hsv\_s=0.7, hsv\_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy\_paste=0.0, cfg=None, tracker=botsort.yaml, save\_dir=runs/detect/train7

* Các lớp và thông số mô hình:

A screenshot of a computer

Description automatically generated

**Hình 23.** Thông tin lớp và thông số

* Thông số mỗi vòng chạy:

A screenshot of a computer

Description automatically generated

**Hình 24.** Thông tin thông số của mỗi vòng chạy

* Model sumary, tốc độ và file chứa mô hình đã train xong “runs/detect/train7”:

A black text on a white background

Description automatically generated

**Hình 25**. Thông tin về model , tốc độ và file chứa

* Thiết lập lại locade của google colab cho phù hợp về “UTF-8”:

A close-up of a computer screen

Description automatically generated

**Hình 26**. Cách thiết lập lại locade

* Bước cuối cùng là predict và visualize kết quả thực nghiệm:

A computer screen shot of a program

Description automatically generated

A close-up of a computer code

Description automatically generated

**Hình 27**. Cách trực quan hóa kết quả thực nghiệm

***Lưu ý :***

Down file dữ liệu dataset đi kèm vào thư mục final trong google drive (drive/MyDrive/final/) thông qua đường link sau đây :

[**https://drive.google.com/file/d/1LE08h0aCKoaF9k18TPsKzbifSLjXb1Vi/view?usp=sharing**](https://drive.google.com/file/d/1LE08h0aCKoaF9k18TPsKzbifSLjXb1Vi/view?usp=sharing)

Nếu như có vấn đề phát sinh chúng ta có thể trực tiếp sử dụng model đã train trong bài báo cáo qua link và trực tiếp đến bước predict+visualize:

[**https://drive.google.com/file/d/11VLGq9M5WvQCPG8BLeot-a3cN0WxZOFU/view?usp=sharing**](https://drive.google.com/file/d/11VLGq9M5WvQCPG8BLeot-a3cN0WxZOFU/view?usp=sharing)

## Kết quả thực nghiệm

Sau đây là 3 trường hợp kết quả thực nghiệm khác nhau mà em lọc ra được.

1. **Kết quả tốt nhất qua ảnh test1.jpg:**

A screenshot of a computer screen

Description automatically generated

**Hình 28**. Kết quả tốt nhất

1. **Kết quả tệ nhất qua ảnh test2.jpg:**

A collage of a street sign

Description automatically generated

**Hình 29**. Kết quả tệ nhất

1. **Kết quả bị nhiễu qua ảnh test3.jpg:**

A road with signs on it

Description automatically generated

**Hình 30**. Kết quả bị nhiễu

**Tài liệu tham khảo**

**Tiếng anh**

[1] "Ultralytics"," Ultralytics Yolov8",2024.

[*https://docs.ultralytics.com/vi#yolo-licenses-how-is-ultralytics-yolo-licensed*](https://docs.ultralytics.com/vi#yolo-licenses-how-is-ultralytics-yolo-licensed)

[2] "Hugging Face","Yolov8",2024.

[*https://huggingface.co/Ultralytics/YOLOv8*](https://huggingface.co/Ultralytics/YOLOv8)

[3]"An Tran Cong, Lu Dien Vuong, Hiep Xuan Huynh","A Model for Real-Time Traffic Signs Recognition Based on the YOLO Algorithm",2019.

[*https://s.net.vn/SDUO*](https://s.net.vn/SDUO)