

Trường Đại học Quy Nhơn

Khoa Công Nghệ Thông Tin



**BÁO CÁO TIỂU LUẬN HỌC PHẦN XỬ LÝ
ẢNH SỐ**

ĐỀ TÀI: NHẬN DIỆN BẢNG SỐ XE

Nhóm Sinh Viên thực hiện:

1. Đặng Gia Thừa
2. Tạ Xuân Kiên
3. Hồ Thị Cẩm Ly
4. Huỳnh Thái Sang

Giảng viên hướng dẫn: TS. Lê Thị Kim Nga

Quy Nhơn, tháng 06 năm 2023

MỞ ĐẦU

Hiện nay, số lượng xe cộ tham gia giao thông trên đường là rất lớn dẫn đến tiêu tốn rất nhiều nhân lực và vật lực cho việc quản lý phương tiện cá nhân trong bãi gửi xe. Nếu không có một công cụ thuận tiện thì việc quản lý phương tiện cá nhân rất mất thời gian, dễ gây nhầm lẫn, thiệt hại cho người sử dụng.

Để giảm tải cho các công việc như thu tiền, bảo hiểm xe, tìm xe cộ trong bãi đỗ xe, trên thế giới đã phát triển công nghệ giám sát tự động đối với các phương tiện giao thông, chính nhờ tính cá nhân của biển số xe mà nó đã trở thành đối tượng chính được sử dụng để nghiên cứu, phát triển trong công nghệ này.

Do đó chúng em muốn chọn đề tài này như bước căn bản trong việc tìm hiểu các công cụ giám sát mạnh hơn như kiểm soát xe lưu thông trên đường hay nhận dạng khuôn mặt, ... đang được thế giới rất chú trọng lúc này.

MỤC LỤC

1. Giới thiệu tổng quan về đề tài	7
1.1 Nhiệm vụ đề tài	7
1.2 ANPR - Automatic Number Plate Recognition	7
1.2.1 ANPR là gì?	7
1.2.2 Thành phần cơ bản của một hệ thống nhận diện biển số xe tự động.....	8
1.2.3 Cách hoạt động của ANPR.....	9
1.2.4 Thực thi Deep Learning trong ANPR	9
1.3 Ngôn ngữ lập trình - Python	10
1.4 Các công cụ hỗ trợ	11
1.4.1 Yolo v3.....	11
1.4.2 EasyOcr (Nhận diện ký tự)	13
1.4.3 Các thư viện và module hỗ trợ khác	14
2. Tổng quan bài toán nhận diện biển số xe	15
2.1 Khái niệm biển số xe.....	15
2.2 Xử lý ảnh và Open CV.....	16
2.3 Hướng giải quyết.....	17
3. Đọc ảnh và xử lý ảnh	17
3.1 Chuyển ảnh xám	17
3.2 Tăng độ tương phản	18
3.2.1 Phép toán hình thái học	18
3.2.2 Tăng độ tương phản	21
3.3 Giảm nhiễu bằng bộ lọc Gauss	22
3.3.1 Nhiễu	22
3.3.2 Bộ lọc Gauss (Gauss filter).....	23
4. Khai thác, phân đoạn và nhận diện biển số:.....	24
4.1 Hàm tính IOU	25
4.2 Loss Function.....	26

4.2.1	Classification loss	26
4.2.2	Localization loss	27
4.2.3	Confidence loss	28
4.2.4	Total loss	28
4.3	Thuật toán Non-Maximum Suppression (NMS).....	28
4.4	Phân đoạn các kí tự có trong biển số xe.....	30
4.5	Nhận diện các kí tự đó rồi đưa về mã ASCII.....	31
5.	Kết quả thực nghiệm	31
5.1	Cách thức đo đạc, thử nghiệm.....	31
5.2	Kết quả thử nghiệm.....	32
6.	Kết luận và hướng phát triển.....	34
6.1	Kết luận	34
6.2	Hướng phát triển	36
7.	Tài liệu tham khảo	38

PHỤ LỤC HÌNH ẢNH

Hình 1 Ứng dụng của ANPR (Nhận diện biển số xe tự động)	8
Hình 2 . ANPR.....	9
Hình 3 ANPR Camera với Deep Learning	10
Hình 4. Logo Python	10
Hình 5. Ngôn ngữ lập trình python	11
Hình 6. EasyOCR	14
Hình 7. OpenCV-Python	14
Hình 8. Matplotlib	15
Hình 9. Numpy	15
Hình 10. Các bước chính trong nhận dạng biển số xe.....	17
Hình 11 Ví dụ về phân tử cấu trúc.....	18
Hình 12 Phép co	19
Hình 13 Phép giãn nở	19
Hình 14 Phép mở	20
Hình 15 Phép đóng	20
Hình 16 Phép Top Hat.....	21
Hình 17 Phép Black Hat	21
Hình 18 Ảnh sau khi tăng độ tương phản.....	22
Hình 19 Nhiễu	22
Hình 20 Ma trận lọc Gauss.....	23
Hình 21 Kết quả sử dụng bộ lọc Gauss	24
Hình 22 Hình trích xuất biển số xe.....	24
Hình 23 nguyên lý hoạt động của yolo.....	25
Hình 24 IOU	26
Hình 25 Vd về Localization loss	27

Hình 26 Trước khi NMS.....	30
Hình 27 Sau khi NMS	30
Hình 28 ví dụ về phân loại bảng số xe	31
Hình 29 Cơ sở dữ liệu các mẫu biển số.....	31
Hình 30 Kết quả thử nghiệm với 1 ảnh	32
Hình 31 kết quả thử nghiệm với 1 ảnh qua từng bước	33
Hình 32 Kết quả thử nghiệm với nhiều ảnh	33
Hình 33 final.....	34

1. Giới thiệu tổng quan về đề tài

Cùng với sự tiến bộ của phát triển khoa học kỹ thuật hiện nay, nhu cầu đi lại của con người ngày càng tăng, lưu lượng giao thông ngày càng lớn. Với số lượng phương tiện giao thông ngày càng lớn và còn không ngừng tăng thì việc quản lý các phương tiện giao thông gặp rất nhiều khó khăn và chính vì thế mà ta cần một hệ thống tự động. Một trong những hệ thống tự động là hệ thống nhận dạng biển số xe tự động. Đây chính là một hệ thống có khả năng “đọc” và “hiểu” các biển số xe một cách tự động mà không đến sự can thiệp của con người. Mục đích của đề tài này chính là tìm hiểu và xây dựng được 1 hệ thống nhận diện biển số xe tự động có triển khai và sử dụng Deep Learning

Nội dung:

- Tìm hiểu về biển số xe và hệ thống nhận dạng biển số xe
- Phát biểu bài toán và hướng giải quyết
- Nghiên cứu một số thuật toán xử lý ảnh và nhận dạng kí tự ứng dụng trong việc nhận dạng biển số xe

1.1 Nhiệm vụ đề tài

Từ nội dung nêu trên, đề tài của chúng em sẽ bao gồm các nhiệm vụ sau:

- Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe.
- Tìm hiểu thông tin về biển số xe và phân loại biển số xe.
- Tìm hiểu các công đoạn chính của bài toán nhận dạng biển số xe gồm 7 khâu chính:
 - Chụp ảnh xe
 - Xử lý ảnh
 - Nhận diện các đối tượng
 - Giảm đối tượng trùng lặp
 - Khai thác biển số
 - Phân đoạn ký tự
 - Nhận dạng ký tự
- Cài đặt thử nghiệm.

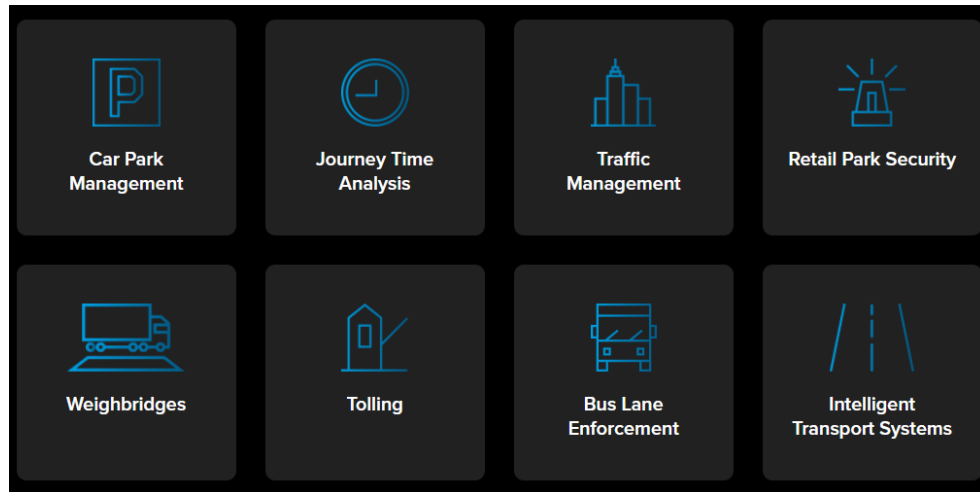
1.2 ANPR - Automatic Number Plate Recognition

1.2.1 ANPR là gì?

Đúng như với tên gọi của nó ANPR là một hệ thống ghi lại biển số xe bằng cách đọc những biển số xe từ camera giám sát giao thông. ANPR là

công nghệ cơ bản dùng để tìm kiếm biển số xe, xe và nó cung cấp thông tin này cho giai đoạn xử lý tiếp theo của máy tính, qua đó thông tin có thể được diễn giải, lưu trữ hoặc đối sánh để tạo một ứng dụng dựa trên ANPR.

Những phần mềm như thế này đã xuất hiện từ năm 1970, nhưng qua ngày tháng phát triển của công nghệ thông tin trong những năm qua, công nghệ cơ bản để nhận diện biển số đã cải tiến, phát triển và cải thiện mạnh mẽ.



Hình 1 Ứng dụng của ANPR (Nhận diện biển số xe tự động)

1.2.2 Thành phần cơ bản của một hệ thống nhận diện biển số xe tự động

1.Computer vision (Thị Giác Máy Tính) và OCR – Optical Character Recognition (Nhận Dạng Ký Tự Quang Học)

2.Phần cứng cung cấp hình ảnh, thường là các camera giám sát chuyên dụng được lắp đặt trên đường và cao tốc

3.Cơ sở dữ liệu để lưu trữ thông tin xe trong trường hợp có bất kỳ sự kiện nào như chạy quá tốc độ, vượt đèn đỏ, v.v. được phát hiện bởi hệ thống ANPR.

Và thành phần quan trọng nhất của một hệ thống ANPR cơ bản chính là thành phần số 1, nó quyết định đến việc một máy tính làm sao để thể đọc được biển số từ video hay hình ảnh, nhanh hay chậm, chính xác hay không chính xác.



Hình 2 . ANPR

1.2.3 Cách hoạt động của ANPR

Nhận diện biển số tự động là 1 quá trình được thực hiện trong 3 bước xử lí:

1. Phát hiện biển số xe:

Trong một đoạn video giao thông, sẽ có rất nhiều chiếc xe đi qua trong từng khoảnh khắc. Nên việc đầu tiên rất quan trọng chính là phát hiện và định vị hóa biển số xe của từng chiếc xe trong mỗi frame. Đây là một vấn đề thiết yếu của phát hiện đối tượng

2. Phân đoạn chữ số:

Ở bước này, biển số được phát hiện và được định vị hóa ở bước trên được phân đoạn thành nhiều vùng để tách biệt từng ký tự trên biển số. Bởi vì những ký tự không được phân đoạn rõ ràng thì hệ thống không thể đọc được số trên biển số.

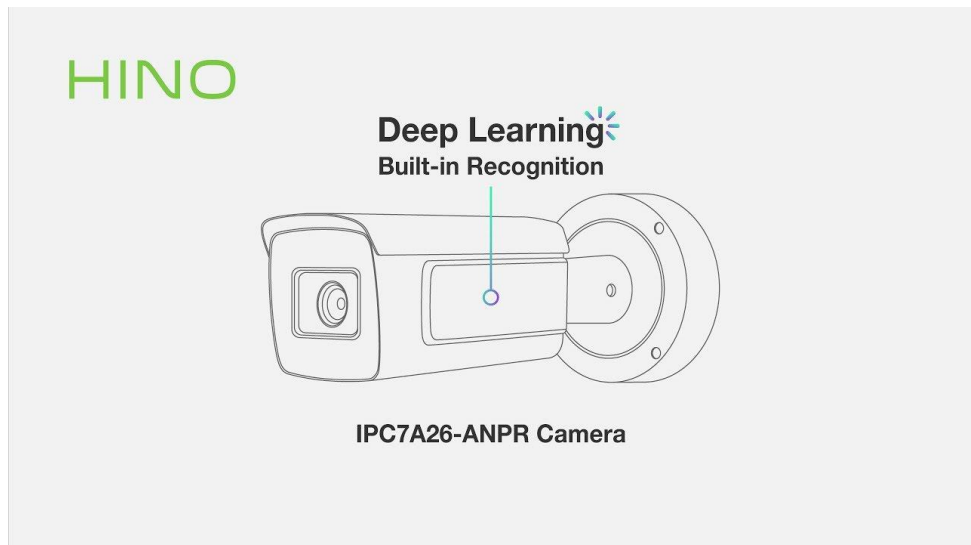
3. Nhận diện chữ số:

Đây là bước cuối cùng, trong đó các ký tự được phân đoạn ở trên được phân loại thành số thực

1.2.4 Thực thi Deep Learning trong ANPR

Hiện tại thì tất cả vấn đề của thị giác máy tính cho thấy những kết quả tốt nhất với kĩ thuật Deep Learning, nó rất chính xác bởi việc sử dụng mạng lưới nơ ron phức hợp trong AI.

Tại thời điểm bây giờ trạng thái mạnh nhất của kĩ thuật Deep learning cho việc thực thi ANPR là YOLO (You Only Look Once).



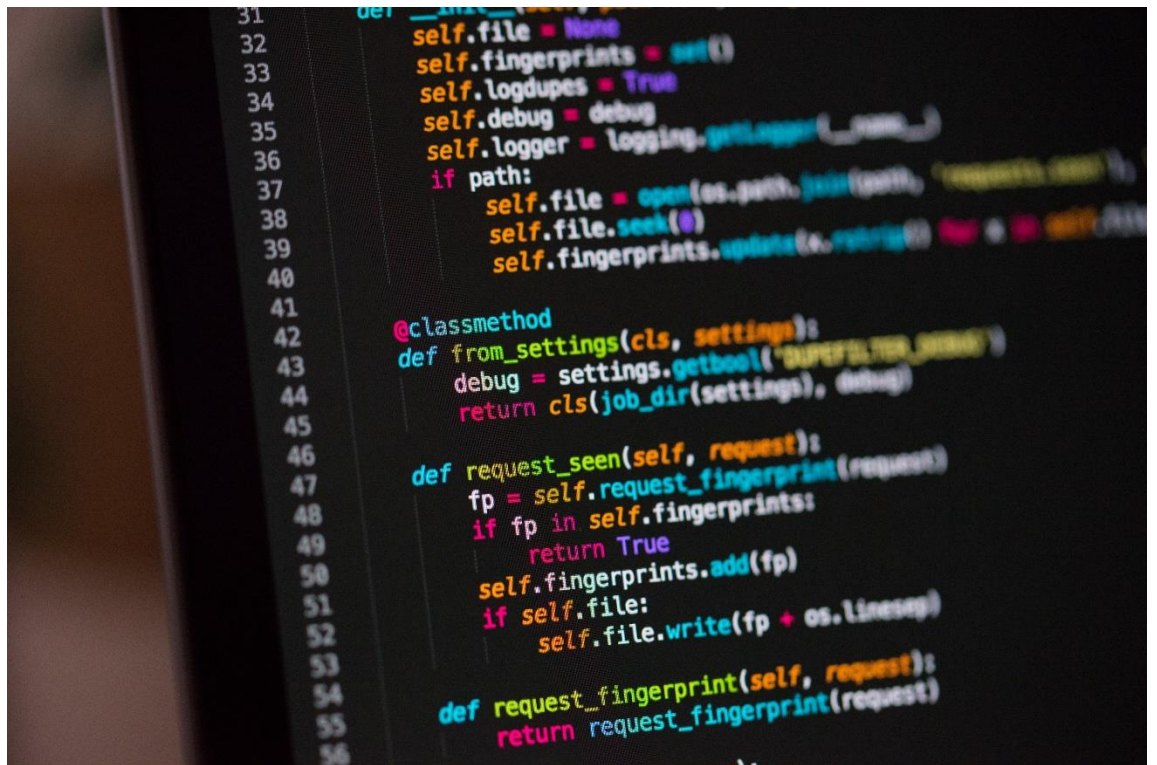
Hình 3 ANPR Camera với Deep Learning

1.3 Ngôn ngữ lập trình - Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.



Hình 4. Logo Python



Hình 5. Ngôn ngữ lập trình python

1.4 Các công cụ hỗ trợ

1.4.1 Yolo v3

1.4.1.1 Yolo v3 là gì?

YOLO (You Only Look Once) là một kiến trúc mạng neural cho việc nhận dạng đối tượng trong ảnh và video. YOLO v3 là phiên bản thứ ba và là một trong những phiên bản phổ biến nhất của YOLO.

YOLO v3 sử dụng một kiến trúc mạng neural convolutional network để phát hiện đối tượng trong hình ảnh. Nó chia ảnh đầu vào thành một lưới ô vuông và dự đoán các hộp giới hạn (bounding boxes) và các lớp đối tượng tương ứng với mỗi ô vuông. Với mỗi ô vuông, YOLO v3 dự đoán các hộp giới hạn và xác suất của các lớp đối tượng khác nhau trong một lần đồng thời.

YOLO v3 cải tiến so với các phiên bản trước bằng cách sử dụng các khối mạng Residual để giảm thiểu hiện tượng mất mát thông tin (information loss) và tăng cường khả năng phát hiện các đối

tượng nhỏ hơn. Nó cũng sử dụng kỹ thuật skip connection để kết hợp thông tin từ các tầng ẩn trung gian, giúp cải thiện khả năng định vị và phân loại đối tượng.

Với kiến trúc mạng đơn giản và khả năng phát hiện nhanh chóng, YOLO v3 đã trở thành một công cụ phổ biến cho nhiều ứng dụng như nhận dạng đối tượng trong ảnh, nhận dạng biển số xe, theo dõi đối tượng trong video, và nhiều ứng dụng khác trong lĩnh vực thị giác máy tính.

1.4.1.2 Cách Yolo v3 hoạt động?

YOLO v3 (You Only Look Once) hoạt động dựa trên kiến trúc mạng neural convolutional network để nhận diện đối tượng trong ảnh.

Quá trình hoạt động của YOLO v3 được tóm tắt như sau:

- Chuẩn bị mạng neural: YOLO v3 sử dụng một mạng neural pre-trained, thường là mạng Darknet-53, để trích xuất các đặc trưng từ ảnh đầu vào. Mạng Darknet-53 bao gồm nhiều tầng convolution và kết nối đầy đủ (fully connected layers) để học các đặc trưng cấp cao của ảnh.
- Tạo lưới ô vuông: Ảnh đầu vào được chia thành một lưới ô vuông (grid), trong đó mỗi ô vuông có kích thước cố định. Số lượng ô vuông phụ thuộc vào kích thước ảnh và độ phân giải mong muốn.
- Dự đoán bounding boxes và xác suất lớp đối tượng: Với mỗi ô vuông trong lưới, YOLO v3 dự đoán các bounding boxes và xác suất tương ứng của các lớp đối tượng. Mỗi bounding box được mô tả bằng tọa độ của các góc (top-left và bottom-right), kích thước và độ tin cậy (confidence) của việc phát hiện đối tượng.

- Loại bỏ bounding boxes không chính xác: Các bounding boxes có độ tin cậy thấp hoặc trùng lặp được loại bỏ để chỉ giữ lại những bounding boxes chính xác nhất.
- Áp dụng ngưỡng xác suất và non-max suppression: Các bounding boxes có xác suất dưới ngưỡng được loại bỏ. Sau đó, áp dụng kỹ thuật non-max suppression để loại bỏ các bounding boxes trùng lặp, chỉ giữ lại bounding box có xác suất cao nhất.
- Đưa ra kết quả: Cuối cùng, YOLO v3 đưa ra kết quả nhận diện đối tượng trên ảnh, bao gồm các bounding boxes và các nhãn (labels) tương ứng với lớp đối tượng được dự đoán.

Qua các bước trên, YOLO v3 cho phép nhận diện nhanh chóng và đồng thời các đối tượng trong ảnh, với khả năng xử lý thời gian thực. Tuy nhiên, để đạt được độ chính xác cao, việc huấn luyện mô hình trên tập dữ liệu đa dạng và tinh chỉnh các siêu tham số là rất quan trọng

1.4.2 EasyOcr (Nhận diện ký tự)

EasyOCR là một package python cho phép chuyển đổi hình ảnh thành văn bản. Cho đến nay, đây là cách dễ nhất để triển khai OCR và có quyền truy cập vào hơn 70 ngôn ngữ bao gồm tiếng Anh, tiếng Trung, tiếng Nhật, tiếng Hàn, tiếng Hindi, nhiều ngôn ngữ khác đang được bổ sung. EasyOCR được tạo ra bởi công ty Jaidev AI. Trong bài viết này, chúng tôi sẽ thảo luận về cách triển khai OCR bằng EasyOCR.

EasyOCR được xây dựng với thư viện Deep Learning Python và Pytorch, có GPU có thể tăng tốc toàn bộ quá trình phát hiện. Phần phát hiện đang sử dụng thuật toán CRAFT và mô hình Nhận dạng là CRNN. Nó bao gồm 3 thành phần chính, trích xuất tính năng, ghi nhãn trình tự (LSTM) và giải

mã (CTC). EasyOCR không có nhiều phụ thuộc vào phần mềm, nó có thể được sử dụng trực tiếp với API của nó.



Hình 6. EasyOCR

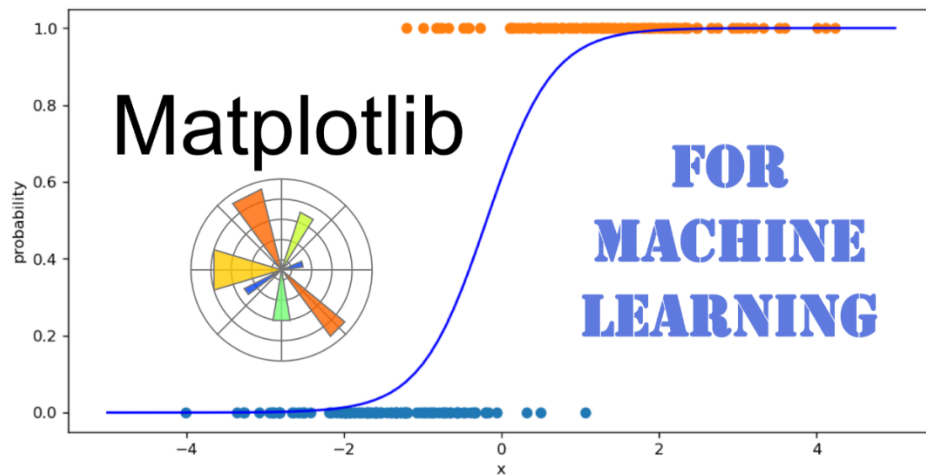
1.4.3 Các thư viện và module hỗ trợ khác

- OpenCV-Python



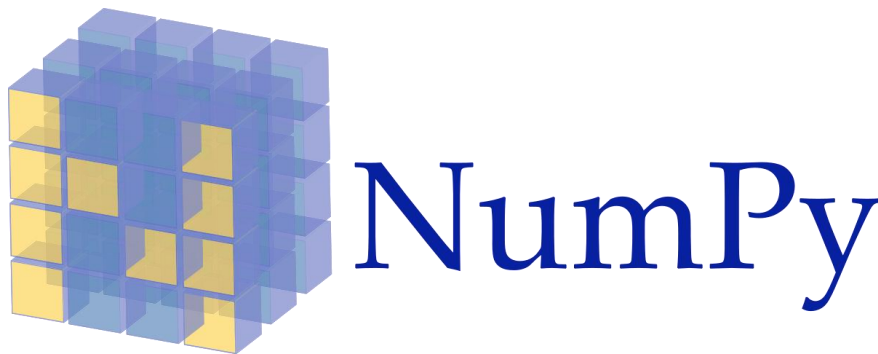
Hình 7. OpenCV-Python

- Matplotlib



Hình 8. Matplotlib

- Numpy



Hình 9. Numpy

2. Tổng quan bài toán nhận diện biển số xe

2.1 Khái niệm biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe) là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim nhôm sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ (biển xe dân sự không dùng các chữ cái I, J, O, Q, W. Chữ R chỉ dùng cho xe rơ-móc, sơ-mi rơ-móc) cho biết: Vùng và địa phương quản lý, các con số cụ thể khi tra trên máy tính còn cho biết danh tính người chủ hay đơn vị đã mua nó, thời gian mua nó phục vụ cho công tác an ninh, đặc biệt trên đó còn có hình Quốc huy Việt Nam đập nổi.

Tiêu chuẩn về kích thước: Ở mỗi nước thường có tiêu chuẩn về kích thước nhất định, còn riêng Việt Nam tỉ lệ kích thước giữa các biển số là gần như giống nhau. Biển số xe có 2 loại, kích thước như sau: Loại biển số dài có chiều cao 110 mm, chiều dài 470 mm; loại biển số ngắn có chiều cao 200 mm, chiều dài 280 mm nên ta sẽ giới hạn tỉ lệ cao/rộng là

$3.5 \leq \text{cao/rộng} \leq 6.5$ (biển một hàng) và $0.8 \leq \text{cao/rộng} \leq 1.5$ (biển hai hàng).

Số lượng kí tự trong biển số xe nằm trong khoảng [7,9]. Chiều cao của chữ và số: 80mm, chiều rộng của chữ và số: 40mm.

Từ những đặc điểm trên ta có thể thiết lập nhưng thông số, điều khiển để lọc chọn những đối tượng phù hợp mà ta cần.

2.2 Xử lý ảnh và Open CV

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế, ...

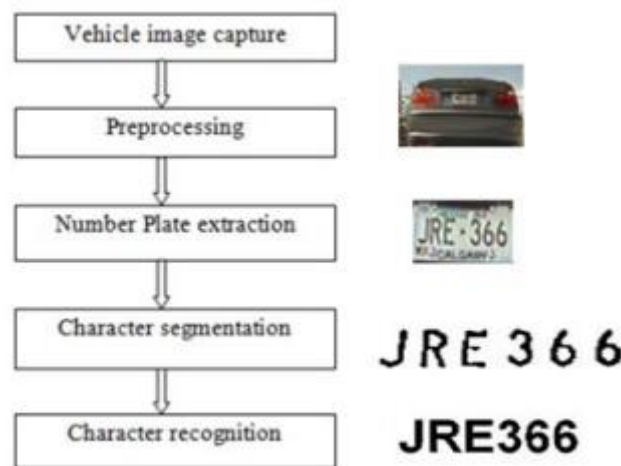
OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOS lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Opencv có rất nhiều ứng dụng như:

- Nhận dạng ảnh
- Xử lý hình ảnh
- Phục hồi hình ảnh/video
- Thực tế ảo
- Các ứng dụng khác

2.3 Hướng giải quyết

Hiện nay trên thế giới đã có rất nhiều cách tiếp cận khác nhau với việc nhận dạng biển số xe, tuy nhiên trong phạm vi đề án này chúng em sẽ giải quyết vấn đề theo 5 bước chính:

1. Chụp ảnh xe
2. Xử lý ảnh
3. Khai thác biển số: nhận diện các đối tượng và giảm đối tượng trùng lặp
4. Phân đoạn các kí tự có trong biển số xe
5. Nhận diện các kí tự đó rồi đưa về mã ASCII



Hình 10. Các bước chính trong nhận dạng biển số xe

3. Đọc ảnh và xử lý ảnh

- Đọc ảnh
- Xử lý ảnh

Quy trình xử lý ảnh:

3.1 Chuyển ảnh xám

Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong dải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để

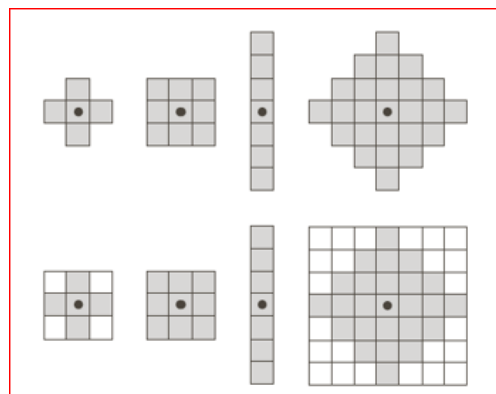
biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý, đơn giản hóa giải thuật nhưng vẫn đảm bảo các tác vụ cần thiết

Ở bài này em sẽ chuyển ảnh xám từ hệ màu HSV thay vì RGB vì với không gian màu HSV ta có ba giá trị chính là: Vùng màu (Hue), độ bão hòa (Saturation), cường độ sáng (Value). Vì lý do đó không gian màu HSV thích nghi tốt hơn đối với sự thay đổi ánh sáng từ môi trường ngoài. Khi chuyển đổi, ảnh xám ta cần là ma trận các giá trị cường độ sáng tách ra từ hệ màu HSV.

3.2 Tăng độ tương phản

3.2.1 Phép toán hình thái học

Hình thái học toán học là một lý thuyết và kỹ thuật để phân tích và xử lý cấu trúc hình học, hình ảnh đầu ra được xác định chủ yếu dựa vào phần tử cấu trúc (structuring elements/kernel)



Hình 11 Ví dụ về phần tử cấu trúc

Hình thái học toán học đã được phát triển cho hình ảnh nhị phân, và sau đó được mở rộng cho ảnh xám,... Đây là một trong những kỹ thuật được áp dụng trong giai đoạn tiền xử lý. Hai phép toán thường dùng là phép giãn nở (Dilation) và phép co (Erosion). Từ hai phép toán cơ bản

này người ta phát triển thành một số phép toán như phép đóng (Closing) và phép mở (Opening) và phép Top Hat, Black Hat.

3.2.1.1 Phép co

Phép toán co có ứng dụng trong việc giảm kích thước của đối tượng, tách rời các đối tượng gần nhau, làm mảnh và tìm xương của đối tượng.



Ảnh gốc



Ảnh sau khi dùng phép co

Hình 12 Phép co

3.2.1.2 Phép giãn nở

Phép toán này có tác dụng làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước (giãn nở ra). Ứng dụng của phép giãn nở là làm cho đối tượng trong ảnh được tăng lên về kích thước, các lỗ nhỏ trong ảnh được lấp đầy, nối liền đường biên ảnh đối với những đoạn rời nhỏ.



Ảnh gốc



Ảnh sau khi dùng phép giãn nở

Hình 13 Phép giãn nở

3.2.1.3 Phép mở

Là thực hiện phép co trước sau đó mới thực hiện phép giãn nở. Phép toán mở được ứng dụng trong việc loại bỏ các phần lõi lõm và làm cho đường bao các đối tượng trong ảnh trở nên mượt mà hơn.



Ảnh gốc



Ảnh sau khi dùng phép mở

Hình 14 Phép mở

3.2.1.4 Phép đóng

Thực hiện phép giãn nở trước sau đó mới thực hiện phép co. Phép toán đóng được dùng trong ứng dụng làm trơn đường bao các đối tượng, lấp đầy các khoảng trống bên trong và loại bỏ những hố nhỏ.



Ảnh gốc



Ảnh sau khi dùng phép đóng

Hình 15 Phép đóng

3.2.1.5 Phép Top Hat

Phép Top Hat là kết quả của phép trừ ảnh của ảnh ban đầu với ảnh sau khi thực hiện phép mở, dùng để làm nổi bật nhưng chi tiết trắng trong nền tối



Ảnh gốc



Ảnh sau khi dùng phép Top Hat

Hình 16 Phép Top Hat

3.2.1.6 Phép Black Hat

Phép Black Hat là kết quả của phép trừ ảnh của ảnh sau khi thực hiện phép đóng với ảnh ban đầu. Dùng làm nổi bật chi tiết tối trong nền trắng.



Ảnh gốc



Ảnh sau khi dùng phép Black Hat

Hình 17 Phép Black Hat

3.2.2 Tăng độ tương phản

Để làm tăng độ tương phản của ảnh số, em sử dụng chủ yếu hai phép Top Hat và Black Hat. Ý tưởng chung là ảnh đầu ra sẽ là ảnh gốc cộng thêm ảnh qua phép Top Hat và trừ đi ảnh qua phép Black Hat. Những chi tiết đã sáng sẽ sáng hơn và những chi tiết tối lại càng tối hơn, từ đó sẽ làm tăng độ tương phản cho ảnh số.



Ảnh gốc



Ảnh sau khi tăng độ tương phản

Hình 18 Ảnh sau khi tăng độ tương phản

3.3 Giảm nhiễu bằng bộ lọc Gauss

3.3.1 Nhiễu

Noise được hiểu cơ bản là các dạng chấm hạt nhỏ phân bố trên hình ảnh. Noise có thể làm biến dạng các chi tiết trong ảnh khiến cho chất lượng ảnh thấp.

Trên thực tế có nhiều loại nhiễu, nhưng người ta thường chia làm ba loại: nhiễu cộng, nhiễu nhân và nhiễu xung. Bản chất của nhiễu thường tương ứng với tần số cao và cơ sở lý thuyết của bộ lọc là chỉ cho những tín hiệu có tần số nhất định đi qua, nên người ta thường sử dụng bộ lọc thông thấp hay trung bình.

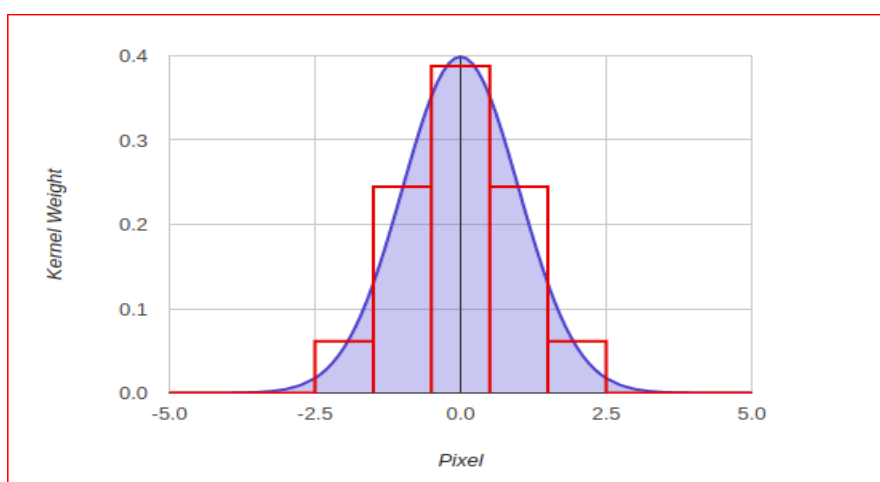


Hình 19 Nhiễu

3.3.2 Bộ lọc Gauss (Gauss filter)

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chap ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).



Hình 20 Ma trận lọc Gauss

Giả sử ảnh là một chiều. Điểm ảnh ở trung tâm sẽ có trọng số lớn nhất. Các điểm ảnh ở càng xa trung tâm sẽ có trọng số giảm dần khi khoảng cách từ chúng tới điểm trung tâm tăng lên. Như vậy điểm càng gần trung tâm sẽ càng đóng góp nhiều hơn vào giá trị điểm trung tâm.



Ảnh gốc

Ảnh sau khi làm mờ, giảm nhiễu

Hình 21 Kết quả sử dụng bộ lọc Gauss

4. Khai thác, phân đoạn và nhận diện biển số:

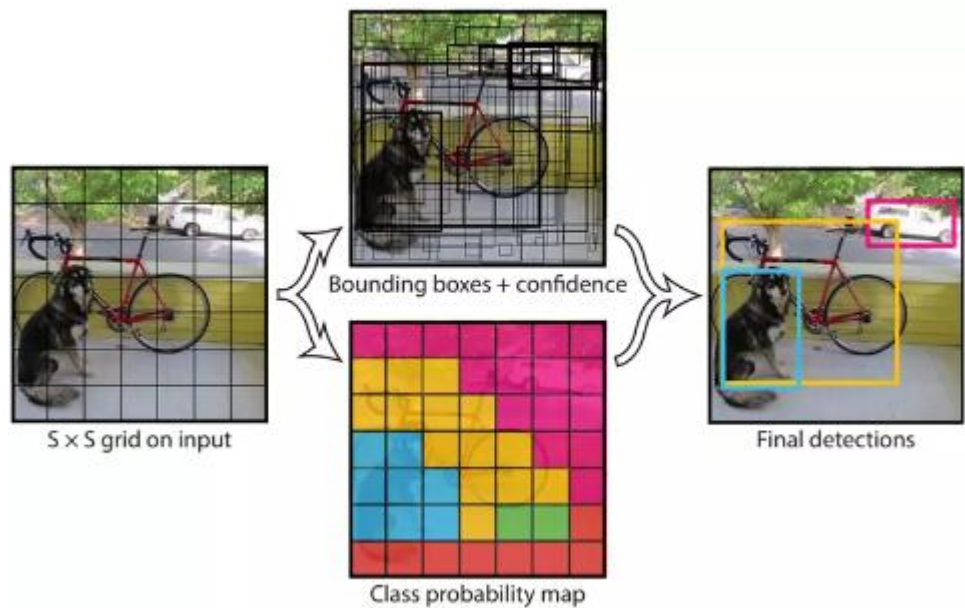
Bước cơ bản trong việc trích xuất biển số là xác định kích thước của biển số và hầu hết các biển số có hình dạng hình chữ nhật. Vì biển số là hình chữ nhật, chúng ta có thể sử dụng hình dạng này như một điểm chính để trích xuất biển số bằng cách loại bỏ phần xe cơ giới



Hình 22 Hình trích xuất biển số xe

Dùng yolov3 để nhận diện đối tượng

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 $S \times S$ ô thường thì sẽ là 3×3 , 7×7 , 9×9 ... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng, mình xin trình bày ở phần trình bày ở phần sau.



Hình 23 nguyên lý hoạt động của yolo


Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước $S \times S \times (5 \times N + M)$ với số lượng tham số mỗi ô là $(5 \times N + M)$ 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 $7 \times 7 \times 7$ ô, mỗi ô cần dự đoán 2 bounding box và 3 object: con chó, ô tô, xe đạp thì output là $7 \times 7 \times 13$, mỗi ô sẽ có 13 tham số, kết quả trả về ($7 \times 7 \times 2 = 98$) bounding box. Chúng ta sẽ cùng giải thích con số $(5 \times N + M)$ được tính như thế nào.

Dự đoán mỗi bounding box gồm 5 thành phần: $(x, y, w, h, \text{prediction})$ 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 $\text{Pr}(\text{Object}) * \text{IOU}(\text{pred}, \text{truth})$ xin trình bày sau. 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 $\text{P}(\text{Object})$, 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 ($\text{P}(\text{chó}|\text{object})$), object2 ($\text{P}(\text{ô tô}|\text{object})$), object3 ($\text{P}(\text{xe đạp}|\text{object})$). 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 bounding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế của mô 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.

4.1 Hàm tính IOU

Trên ta có đề cập prediction được định nghĩa $\text{Pr}(\text{Object}) * \text{IOU}(\text{pred}, \text{truth})$, ta sẽ làm rõ hơn $\text{IOU}(\text{pred}, \text{truth})$ là gì. IOU (INTERSECTION OVER UNION)

là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể. IOU được tính bằng

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


Hình 24 IOU

Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với growth-truth bounding box, còn Area of Union là diện tích phần hợp giữa predicted bounding box với growth-truth bounding box. Những bounding box được đánh nhãn bằng tay trong tập training set và test set. Nếu $\text{IOU} > 0.5$ thì prediction được đánh giá là tốt.

4.2 Loss Function

Hàm lỗi trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính. Cụ thể hơn nó là tổng độ lỗi của 3 thành phần con sau :

- Độ lỗi của việc dự đoán loại nhãn của object - Classification loss
- Độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của boundary box (x, y, w, h) - Localization loss
- Độ lỗi của việc dự đoán bounding box đó chứa object so với nhãn thực tế tại ô vuông đó - Confidence loss

4.2.1 Classification loss

Classification loss - độ lỗi của việc dự đoán loại nhãn của object, hàm lỗi này chỉ tính trên những ô vuông có xuất hiện object, còn những ô vuông khác ta không quan tâm. Classification loss được tính bằng công

thứ

sau:

$$L_{classification} = \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2$$

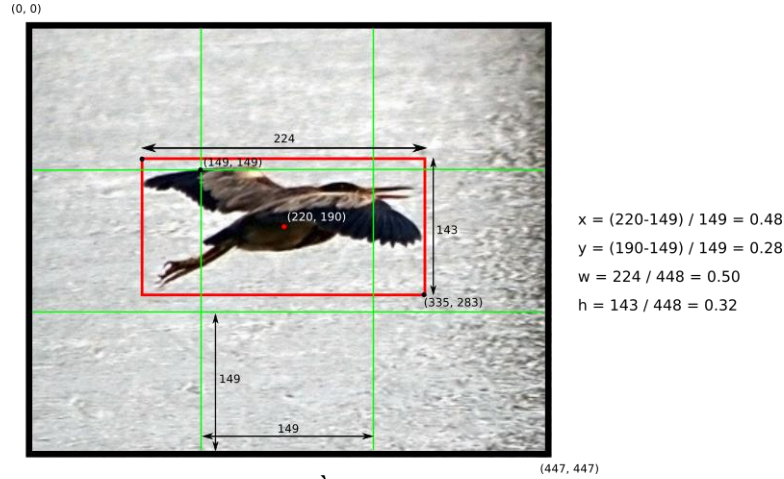
Trong đó:

\mathbb{I}_i^{obj} : bằng 1 nếu ô vuông đang xét có object ngược lại bằng 0

$\hat{p}_i(c)$: là xác suất có điều của lớp c tại ô vuông tương ứng mà mô hình dự đoán

4.2.2 Localization loss

Localization loss là hàm lỗi dùng để tính giá trị lỗi cho boundary box được dự đoán bao gồm tọa độ tâm, chiều rộng, chiều cao của so với vị trí thực tế từ dữ liệu huấn luyện của mô hình. Lưu ý rằng chúng ta không nên tính giá trị hàm lỗi này trực tiếp từ kích thước ảnh thực tế mà cần phải chuẩn hóa về $[0, 1]$ so với tâm của bounding box. Việc chuẩn hóa này kích thước này giúp cho mô hình dự đoán nhanh hơn và chính xác hơn so với để giá trị mặc định của ảnh. Hãy cùng xem một ví dụ:



Hình 25 Vd về Localization loss

Giá trị hàm Localization loss được tính trên tổng giá trị lỗi dự đoán tọa độ tâm (x, y) và (w, h) của predicted bounding box với ground-truth bounding box. Tại mỗi ô có chứa object, ta chọn 1 boundary box có IOU (Intersect over union) tốt nhất, rồi sau đó tính độ lỗi theo các boundary box này. Giá trị hàm lỗi dự đoán tọa độ tâm (x, y) của predicted bounding box và (\hat{x} , \hat{y}) là tọa độ tâm của truth bounding box được tính như sau:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

Giá trị hàm lỗi dự đoán (w, h) của predicted bounding box so với truth bounding box được tính như sau :

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

Với ví dụ trên thì $S = 7$, $B = 2$, còn λ_{coord} là trọng số thành phần trong paper gốc tác giả lấy giá trị là 5

4.2.3 Confidence loss

Confidence loss là độ lỗi giữa dự đoán boundary box đó chứa object so với nhãn thực tế tại ô vuông đó. Độ lỗi này tính trên cả những ô vuông chứa object và không chứa object.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

Với ví dụ trên thì $S = 7$, $B = 2$, còn $\lambda_{noobject}$ là trọng số thành phần trong paper gốc tác giả lấy giá trị là 0.5. Đối với các hộp j của ô thứ i nếu xuất hiện object thì $C_i = 1$ và ngược lại

4.2.4 Total loss

Tổng lại chúng ta có hàm lỗi là tổng của 3 hàm lỗi trên :

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

4.3 Thuật toán Non-Maximum Suppression (NMS)

Áp dụng Non-Maximum Suppression (NMS) để loại bỏ các bounding box trùng lặp

Intersection Over Union (IoU)

Nội dung thuật toán

Input:

Một mảng các bounding box, mỗi box sẽ có dạng $(x1, y1, x2, y2, c)$ trong đó:

- $(x1, y1)$ và $(x2, y2)$ lần lượt là tọa độ điểm top-left và bottom-right của bounding box
- c là confidence score tương ứng với box đó, được trả về từ mô hình object detection.

Giá trị ngưỡng IOU.

Output:

Một mảng các bounding box sau khi đã được loại bỏ đi các bounding box dư thừa.

Chi tiết thuật toán

Các ký hiệu:

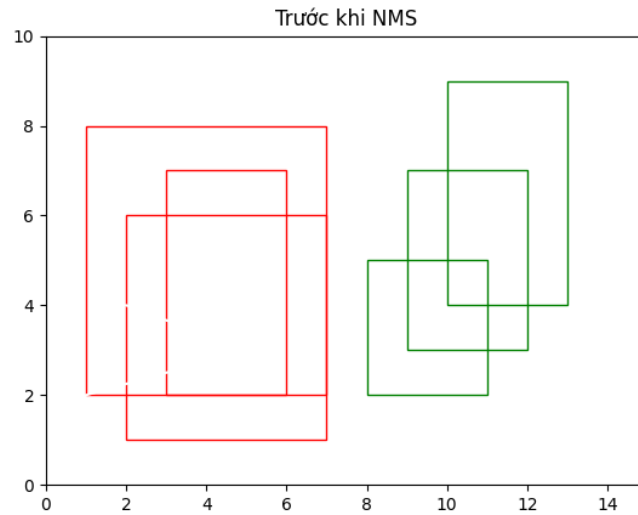
- S : bounding box đang xét
- P : Tập các box đầu vào của thuật toán
- $thresh_iou$: Ngưỡng IoU để loại bỏ các box thừa
- $keep$: Tập các box sau khi đã loại bỏ các box thừa

Thuật toán bao gồm 3 bước:

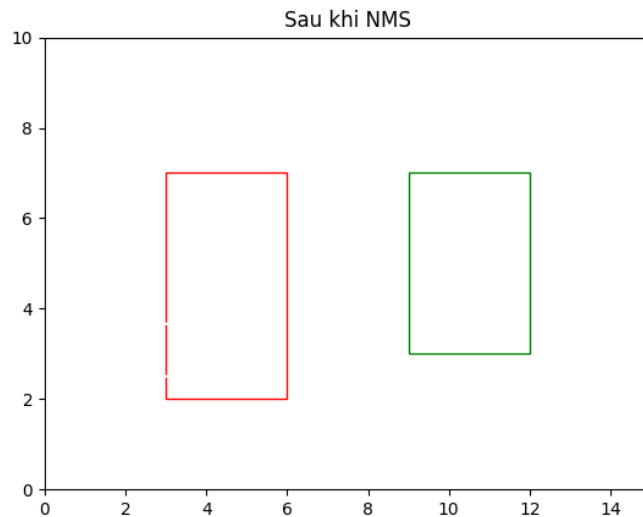
- Bước 1: Chọn box S có confidence score cao nhất trong tập P , loại bỏ box đó ra khỏi tập P và thêm box đó vào tập $keep$.
- Bước 2 Thực hiện tính toán IOU giữa box S vừa lấy ra ở bước 1 với toàn bộ các box còn lại trong tập P . Nếu có box nào trong P có IOU với box S đang xét mà lớn hơn ngưỡng $thresh_iou$ thì loại bỏ box đó ra khỏi P
- Bước 3 Lặp lại bước 1 cho đến khi P không còn box nào.

Sau khi kết thúc thuật toán thì $keep$ chứa toàn bộ những box sau khi đã loại bớt các box thừa.

Nhìn vào các bước của thuật toán trên thì các bạn cũng có thể nhìn thấy việc loại bỏ đi các box chỉ phụ thuộc vào giá trị ngưỡng IoU $thresh_iou$. Do đó việc lựa chọn giá trị ngưỡng này sẽ ảnh hưởng đến hiệu suất của mô hình object detection



Hình 26 Trước khi NMS

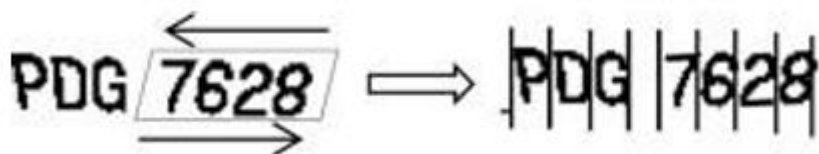


Hình 27 Sau khi NMS

4.4 Phân đoạn các kí tự có trong biển số xe

Phân đoạn ký tự là một quá trình nhằm phân tách một hình ảnh của một chuỗi ký tự thành các hình ảnh con của các ký tự riêng lẻ. Đây là một trong những quyết định trong quá trình nhận dạng ký tự quang học (OCR). Quyết định của nó, rằng một mẫu được cô lập từ hình ảnh là một ký tự (hoặc một đơn vị có thể nhận dạng khác), có thể đúng hoặc sai. Hình ảnh được tìm kiếm để tìm bất kỳ pixel nào thỏa mãn yêu cầu. Mỗi khi gặp một pixel như vậy, các hàng xóm

của nó được kiểm tra và nếu bất kỳ hàng xóm nào cũng tương tự với các tiêu chí, cả hai pixel đều được coi là thuộc về cùng một khu vực. Chúng ta thu được hình ảnh ký tự và số riêng lẻ bằng cách sử dụng kỹ thuật quét theo chiều dọc và ngang.



Hình 28 ví dụ về phân loại bảng số xe

4.5. Nhận diện các ký tự đó rồi đưa về mã ASCII

Đây là giai đoạn quan trọng và cơ bản nhất của hệ thống ANPR. Nó trình bày các kỹ thuật được sử dụng để sắp xếp và nhận dạng các ký tự riêng lẻ. Việc phân loại dựa trên các đặc trưng đã được trích xuất. Nhận dạng ký tự sử dụng kỹ thuật nhận dạng ký tự quang học (OCR) để so sánh từng ký tự riêng lẻ với cơ sở dữ liệu chữ cái và số. OCR thực sự sử dụng phương pháp tương quan để so khớp từng ký tự riêng lẻ và cuối cùng, số được nhận dạng và lưu trữ dưới dạng chuỗi trong một biến. Ký tự sau đó được so sánh với cơ sở dữ liệu quyền sở hữu phương tiện. Các kết quả sau đó được cung cấp dựa trên kết quả của quá trình so sánh. Sẽ tồn tại các mẫu cho tất cả các ký tự từ A-Z và 0-9.



Hình 29 Cơ sở dữ liệu các mẫu biến số

5. Kết quả thực nghiệm

5.1 Cách thức đo đạc, thử nghiệm

Thiết bị thử nghiệm:

CPU: i7 10750h

GPU: GTX-1660ti-maxq

RAM: 16GB

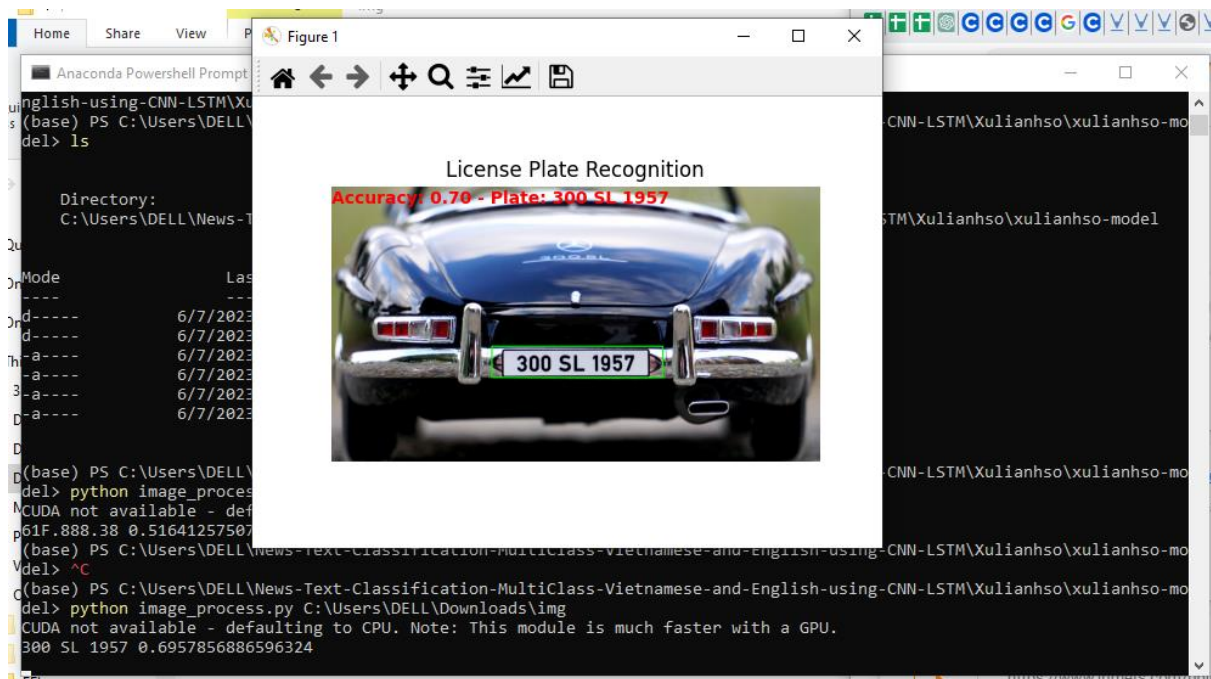
OS: WINDOW 11

5.2 Kết quả thử nghiệm

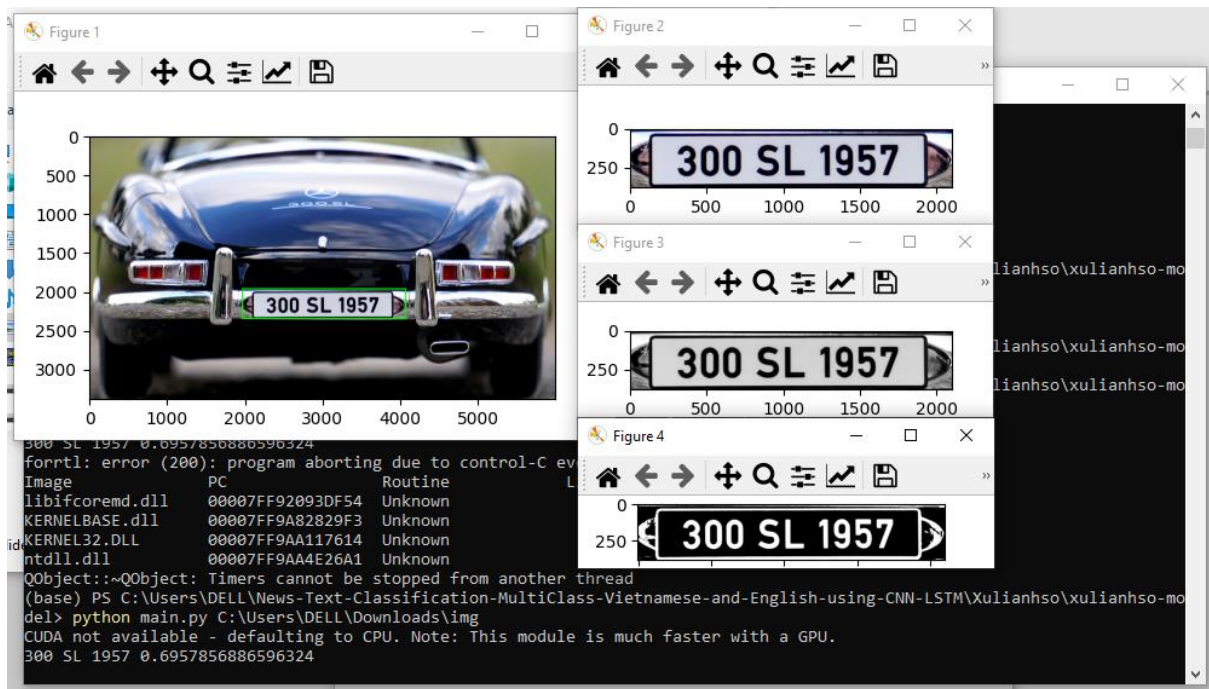
Input: 1 thư mục chứa ảnh

Output: in biển số xe, tỉ lệ chính xác và hình ảnh được đánh dấu vị trí biển số

Trường hợp thư mục chứa 1 ảnh:

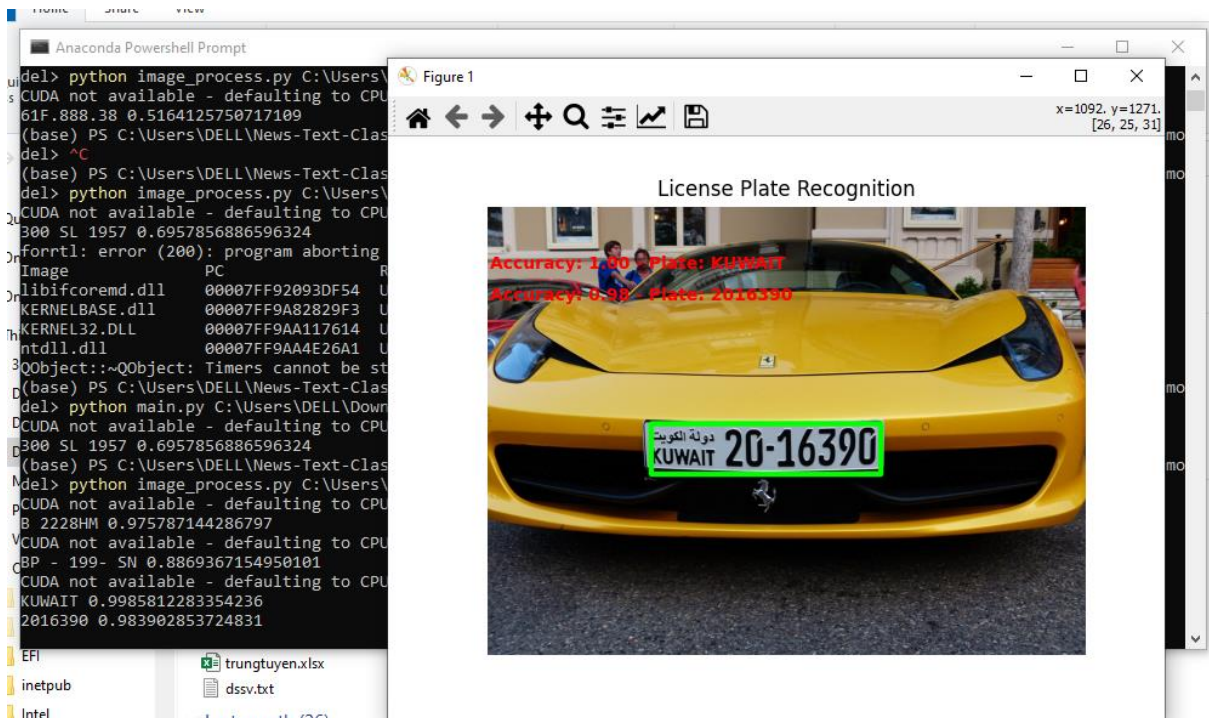


Hình 30 Kết quả thử nghiệm với 1 ảnh



Hình 31 kết quả thử nghiệm với 1 ảnh qua từng bước

Trường hợp thư mục chứa nhiều ảnh:



Hình 32 Kết quả thử nghiệm với nhiều ảnh

```

del> python image_process.py C:\Users\DELL\Downloads\img
CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
B 2228HM 0.975787144286797
h CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
3BP - 199- SN 0.8869367154950101
D CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
KUWAIT 0.9985812283354236
D2016390 0.983902853724831
D CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
AG 397072 0.48555993400559
Graf Automobile AG 0.9212084654342548
D CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
WPX 568 0.9966471236579856
VTOVEKS BIL - Hela vigen 0.6405312686208005
C CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
ARL 0.9996072919413483
333 0.9997927004264573
D CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
300 SL 1957 0.9997915221793378

```

Hình 33 final

6. Kết luận và hướng phát triển

6.1 Kết luận

Từ những kết quả thu được ở chương trên em nhận thấy phương pháp nhận diện biển số xe bằng xử lý ảnh dùng mạng CNN và thuật toán NMS có những ưu và khuyết điểm sau:

Ưu điểm:

- Tính chính xác cao: Sử dụng mạng CNN cho phép trích xuất đặc trưng chính xác từ ảnh và nhận diện đối tượng với độ chính xác cao.
- Xử lý thời gian thực: Kiến trúc YOLO v3 cho phép nhận diện nhanh chóng và đồng thời các đối tượng trong ảnh, với khả năng xử lý thời gian thực. Điều này hữu ích trong các ứng dụng yêu cầu tương tác và đáp ứng nhanh, như theo dõi xe, giám sát giao thông, hay quản lý bãi đỗ xe.
- Kiến trúc đơn giản: YOLO v3 có kiến trúc đơn giản và dễ hiểu, với một mạng neural duy nhất để thực hiện cả quá trình nhận diện và phân loại.
- Xử lý đa đối tượng: YOLO v3 có khả năng nhận diện nhiều đối tượng cùng một lúc trên cùng một ảnh, không cần phải thực hiện nhiều lần quét ảnh.

Khuyết điểm:

- Phụ thuộc vào chất lượng ảnh: Độ chính xác của phương pháp này có thể bị ảnh hưởng bởi chất lượng ảnh đầu vào. Ảnh có độ phân giải thấp, nhiễu hoặc bị mờ có thể làm giảm khả năng nhận diện chính xác.
- Khả năng nhận diện trong điều kiện môi trường khác nhau: Phương pháp này có thể gặp khó khăn trong việc nhận diện biển số xe trong các điều kiện môi trường khác nhau, chẳng hạn như ánh sáng yếu, trời mưa hoặc tình huống mờ mờ.
- Độ phức tạp của thuật toán: Phương pháp này yêu cầu tính toán phức tạp, đặc biệt là khi áp dụng mạng CNN trên các ảnh có độ phân giải cao. Điều này có thể đòi hỏi tài nguyên tính toán mạnh mẽ để đảm bảo hiệu suất và thời gian xử lý.
- Cần tập dữ liệu huấn luyện lớn: Để đạt được độ chính xác cao, phương pháp này yêu cầu một tập dữ liệu huấn luyện lớn.
- Phụ thuộc vào việc chuẩn bị dữ liệu huấn luyện: Để đạt được hiệu suất tốt, phương pháp này đòi hỏi một tập dữ liệu huấn luyện đủ lớn và đa dạng, bao gồm các biển số xe có các biến thể về kích thước, font chữ, màu sắc và điều kiện ánh sáng khác nhau. Việc chuẩn bị và thu thập dữ liệu huấn luyện phù hợp có thể tốn nhiều thời gian và công sức.
- Độ tin cậy của kết quả: Mặc dù phương pháp này có độ chính xác cao, nhưng nó cũng có khả năng phát sinh các sai sót và nhận dạng sai với các biển số xe có hình dạng, màu sắc, font chữ không thông thường. Điều này có thể gây ra nhầm lẫn hoặc bỏ sót trong việc nhận diện biển số xe.
- Khả năng mở rộng và đa dạng hóa: Phương pháp này có thể đối mặt với khó khăn trong việc mở rộng và đa dạng hóa để nhận diện biển số xe ở các quốc gia khác nhau với các quy tắc và đặc điểm biển số khác nhau. Cần có sự tùy chỉnh và thích ứng phương pháp để phù hợp với từng quốc gia hoặc khu vực cụ thể.
- Sự ảnh hưởng của điều kiện môi trường: Phương pháp này có thể không hiệu quả trong việc nhận diện biển số xe trong các điều kiện môi trường khắc nghiệt, như bụi, sương mù hoặc đèn pha sáng chói. Điều này có thể làm giảm độ chính xác và tin cậy của phương pháp.

Tóm lại, phương pháp nhận diện biển số xe bằng xử lý ảnh mạng CNN và thuật toán NMS có nhiều ưu điểm như độ chính xác cao và khả năng xử lý thời gian thực, nhưng cũng có nhược điểm như phụ thuộc vào chất lượng ảnh, khả năng nhận diện trong điều kiện môi trường khác nhau và độ phức tạp của thuật toán. Việc hiểu và đánh giá các ưu và khuyết điểm này là quan trọng để áp dụng phương pháp một cách hiệu quả và tối ưu hóa trong các ứng dụng thực tế. Cần xem xét các yếu tố này để đảm bảo phương pháp nhận diện biển số xe đáp ứng được yêu cầu và điều kiện cụ thể trong môi trường áp dụng. Ngoài ra, việc kết hợp với các phương pháp và thuật toán khác như xử lý ảnh, xử lý văn bản, hay các phương pháp học sâu khác có thể cải thiện độ chính xác và khả năng nhận diện của hệ thống.

6.2 Hướng phát triển

Có một số hướng phát triển tiềm năng để cải tiến phương pháp nhận diện biển số xe bằng xử lý ảnh mạng CNN và thuật toán NMS:

- Tăng độ chính xác: Cải thiện độ chính xác của phương pháp bằng cách tăng cường mô hình mạng CNN. Có thể sử dụng các kiến trúc mạng CNN sâu hơn, kết hợp với các phương pháp tăng cường đặc trưng như skip connections, residual connections, hoặc attention mechanisms để nâng cao khả năng trích xuất đặc trưng và nhận diện chính xác.
- Xử lý dữ liệu đa dạng hơn: Mở rộng tập dữ liệu huấn luyện bằng cách thu thập nhiều biển số xe từ các quốc gia, khu vực khác nhau và trong các điều kiện ánh sáng, môi trường khác nhau. Điều này giúp mô hình học được đa dạng về hình dạng, màu sắc và font chữ của biển số xe, từ đó cải thiện khả năng nhận diện trong các tình huống đa dạng.
- Tối ưu hóa tính toán: Nghiên cứu các phương pháp tối ưu hóa tính toán để giảm chi phí tính toán của mô hình, đồng thời đảm bảo hiệu suất và thời gian xử lý nhanh chóng. Có thể sử dụng các kỹ thuật như pruning, quantization, hoặc sử dụng phần cứng đặc specialized để tăng tốc tính toán.

- Kết hợp với phương pháp xử lý văn bản: Kết hợp phương pháp nhận diện biển số xe với phương pháp xử lý văn bản để trích xuất và nhận diện số và ký tự trên biển số xe. Việc kết hợp này giúp cải thiện độ chính xác và tin cậy của hệ thống.
- Tích hợp học sâu: Sử dụng các phương pháp học sâu khác như mạng LSTM hoặc Transformer để mô hình hóa ngữ cảnh và mối quan hệ giữa các ký tự trên biển số xe. Điều này có thể cải thiện khả năng nhận diện trong các trường hợp biển số xe bị che khuất hoặc có hành vi phức tạp.
- Đa nhiệm hóa: Nghiên cứu việc mở rộng phương pháp và quá trình để nhận diện và phân loại không chỉ biển số xe mà còn các thông tin khác liên quan đến xe, như hãng xe, màu sắc, loại xe, hay trạng thái của xe. Điều này giúp mở rộng ứng dụng của phương pháp và tăng tính đa dạng trong việc phân loại các thông tin liên quan đến xe.
- Tăng cường học máy: Áp dụng kỹ thuật tăng cường học máy (reinforcement learning) để cải thiện khả năng nhận diện và điều chỉnh các thông số trong mô hình như threshold, phạm vi ánh sáng, hoặc độ chính xác. Việc này giúp mô hình tự điều chỉnh và cải thiện kết quả nhận diện theo thời gian.
- Ứng dụng trong thực tế: Tiến hành nghiên cứu và thử nghiệm phương pháp trên các hệ thống thực tế, như hệ thống giám sát giao thông, hệ thống an ninh hay hệ thống quản lý đỗ xe. Từ đó, thu thập phản hồi từ người dùng và tối ưu hóa phương pháp để đáp ứng tốt nhất các yêu cầu và điều kiện trong thực tế.

Tổng kết, các hướng phát triển trên giúp nâng cao độ chính xác, khả năng nhận diện và tính ứng dụng của phương pháp nhận diện biển số xe bằng xử lý ảnh mạng CNN và thuật toán NMS. Việc tiếp tục nghiên cứu và phát triển trong các lĩnh vực trên sẽ đóng góp vào việc cải thiện hiệu suất và độ tin cậy của hệ thống nhận diện biển số xe trong thực tế.

7. Tài liệu tham khảo

- [1] G. D. Yeshwant, S. Maiti, and P. B. Borole, “Automatic Number Plate Recognition System (ANPR System),” Int. J. Eng. Res., vol. 3, no. 7, p. 5, 2014, [Online]. Available: <https://www.ijert.org/research/automatic-number-plate-recognition-system-anpr-system-IJERTV3IS071132.pdf>.
- [2] Tìm hiểu và triển khai thuật toán Non Maximum Suppression Link: <https://viblo.asia/p/tim-hieu-va-trien-khai-thuat-toan-non-maximum-suppression-bJzKmr66Z9N>
- [3] Tìm hiểu về YOLO trong bài toán real-time object detection Link: <https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P>
- [4] AUTOMATIC NUMBER PLATE RECOGNITION USING TENSORFLOW AND EASYOCR Dr. Vishwanath Burkpalli*1, Abhishek Joshi*2, Abhishek B Warad*3, Akash Patil*4 .Link: https://www.irjmets.com/uploadedfiles/paper/issue_9_september_2022/29740/final/fin_irjmets1662817354.pdf