

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



BÁO CÁO ĐỒ ÁN MÔN HỌC NHẬN DẠNG
ĐỀ TÀI: LICENSE PLATE RECOGNITION

Giảng viên: Đỗ Văn Tiến

Lớp: CS419.N21.KHCL

Sinh viên thực hiện:	Trần Minh Phúc	20521782
	Nguyễn Thanh Phúc	20521769
	Nguyễn Minh Nhật	20521708
	Lê Thế Tuấn	20522113

Ho Chi Minh 07/2023

Link github project: <https://github.com/phucnt2002/CS338>

BẢNG PHÂN CÔNG VÀ ĐÁNH GIÁ

STT	MSSV	Họ và tên	Phân công nhiệm vụ	Tỉ lệ hoàn thành
1	2052182	Trần Minh Phúc	Thuyết trình: <ul style="list-style-type: none">Đánh giá mô hìnhTrả lời câu hỏi Viết báo cáo: <ul style="list-style-type: none">Phần I: Giới thiệuPhần IV: Đánh giá mô hìnhPhần VII: Deploy model Cài đặt code: <ul style="list-style-type: none">Deploy model	100%
2	20522113	Lê Thế Tuấn	Thuyết trình: <ul style="list-style-type: none">Optical character regonitionTrả lời câu hỏi Viết báo cáo: <ul style="list-style-type: none">Phần V: Optical character regonition	100%
3	20521769	Nguyễn Thanh Phúc	Thuyết trình: <ul style="list-style-type: none">Deploy modelTrả lời câu hỏi Viết báo cáo: <ul style="list-style-type: none">Phần VII: Deploy modelPhần VIII: Kết luậnPhần IX: Hướng phát triển Cài đặt code: <ul style="list-style-type: none">Yolo v5Deploy model	100%

4	20521708	Nguyễn Minh Nhật	Thuyết trình: <ul style="list-style-type: none"> • Yolo v5 • Trả lời câu hỏi Viết báo cáo: <ul style="list-style-type: none"> • Phần II: Mô tả và xây dựng bộ dữ liệu • Phần III: Các nghiên cứu liên quan • Phần IV: Yolo v5 Cài đặt code: <ul style="list-style-type: none"> • Yolo v5 	100%
---	----------	------------------	---	------

MỤC LỤC

I. TỔNG QUAN	1
1. Giới thiệu về đề án	1
2. Mục tiêu đề án	2
3. Các bước tiến hành	2
II. MÔ TẢ VÀ XÂY DỰNG BỘ DỮ LIỆU	3
1. Mô tả bộ dữ liệu	3
2. Xử lý bộ dữ liệu	3
III. CÁC NGHIÊN CỨU LIÊN QUAN	5
1. Phương pháp nhận dạng biển số xe dựa trên kiến trúc mạng LPR-SSD	5
2. Nhận dạng biển số xe tự động dựa trên artificial neural networks (ANN)	5
3. Nhận dạng biển số xe tự động dựa trên Convolutional Neural Network (CNN)	6
IV. YOLO V5	7
1. Giới thiệu	7
2. Train model	7
V. OPTICAL CHARACTER RECOGNITION	10
1. Giới thiệu	10
2. Các bước thực hiện bài toán OCR	10
2.1. Preprocessing	10
2.2. Text Detection	12
2.3. Text Recognition	13
3. Ưu điểm và Nhược điểm	14
4. Ứng dụng của OCR	14
VI. ĐÁNH GIÁ MÔ HÌNH	17
1. Mô hình Yolo v5	17
2. Mô hình OCR	19
VII. DEPLOY MODEL	20
1. Giới thiệu	20
2. Workflow	21
VIII. KẾT LUẬN	24

IX. HƯỚNG PHÁT TRIỂN	24
REFERENCE.....	26

I. TỔNG QUAN

1. Giới thiệu về đề án

Trong đời sống hiện đại, phương tiện giao thông đóng vai trò rất quan trọng trong việc di chuyển và vận chuyển hàng hóa. Tuy nhiên, việc quản lý xe cộ cũng đặt ra nhiều thách thức cho các cơ quan chức năng. Biển số xe là một yếu tố quan trọng trong việc giám sát giao thông đường bộ và là cách để xác định chính xác các phương tiện giao thông.

Trong thời đại công nghệ thông tin ngày nay, việc nhận dạng và xử lý thông tin từ hình ảnh đã trở thành một lĩnh vực quan trọng và hứa hẹn. Trong đề án này, ta tập trung vào việc phát triển một hệ thống nhận dạng biển số xe tự động.

Đề án sẽ bao gồm các bước chính như xử lý ảnh tiền xử lý để làm sạch và cải thiện chất lượng ảnh, trích xuất đặc trưng từ ảnh biển số, áp dụng thuật toán YOLOv5 và OCR để tìm ra kết quả nhận dạng chính xác nhất.

Kết quả mong đợi từ đề án này là một hệ thống nhận dạng biển số xe có khả năng hoạt động chính xác và ổn định trong các tình huống thực tế, bao gồm ánh sáng yếu, góc chụp khác nhau và biến đổi về màu sắc và kích thước của biển số xe.

Ứng dụng của hệ thống có thể sử dụng trong các lĩnh vực như:

- Giám sát giao thông
- Quản lý đỗ xe
- An ninh



2. Mục tiêu đề án

Mục tiêu chính của đề án là xây dựng một hệ thống nhận dạng biển số xe hiệu quả và chính xác. Nhóm em sử dụng một số phương pháp tiên tiến trong lĩnh vực xử lý ảnh và học máy để nhận dạng và trích xuất thông tin từ hình ảnh biển số xe. Qua đó, hệ thống sẽ tự động phát hiện và nhận dạng biển số xe từ hình ảnh chụp hoặc video và trả về kết quả là nội dung của biển số xe.

3. Các bước tiến hành

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

- Thu thập và tổng hợp dữ liệu: Nhóm đã thu thập một số lượng lớn hình ảnh chứa các biển số xe từ các bộ dataset công khai trên Roboflow để huấn luyện và kiểm thử hệ thống.
- Chuẩn bị dữ liệu:
 - Gán bounding box cho từng hình ảnh trong bộ dữ liệu.
 - Tiền xử lý dữ liệu.
 - Tăng cường dữ liệu.
- Chia bộ dữ liệu thành các tập train, validate, test.
- Huấn luyện mô hình: Sử dụng một mô hình CNN (Convolutional Neural Network) YOLOv5 để huấn luyện cho việc xác định biển số xe trên hình ảnh hoặc video.
- Kiểm thử và hiệu chỉnh: Sau khi huấn luyện, tiến hành kiểm thử và hiệu chỉnh mô hình để đạt được độ chính xác tốt nhất.
- Nhận dạng biển số xe: Sau khi xác định được vị trí của biển số xe trên hình ảnh hoặc video, sử dụng thuật toán OCR để đọc nội dung của biển số xe.

II. MÔ TẢ VÀ XÂY DỰNG BỘ DỮ LIỆU

1. Mô tả bộ dữ liệu

Hình ảnh các biển số xe máy, xe ô tô từ nhiều góc chụp và điều kiện ánh sáng khác nhau được tổng hợp từ nhiều nguồn dataset công khai trên [Roboflow](#)

Bộ dữ liệu gốc: 400 hình ảnh

Bộ dữ liệu sau khi tăng cường: 882 hình ảnh

Một số hình ảnh trong tập dữ liệu:



2. Xử lý bộ dữ liệu

- Tiến hành gán nhãn các biển số cho từng hình ảnh
- Tiền xử lý dữ liệu
 - Auto - Orient
 - Resize to 704x704
- Tăng cường dữ liệu
 - Rotation $\pm 15^\circ$
 - Shear $\pm 15^\circ$ Horizontal and Vertical
 - Saturation $\pm 25^\circ$
 - Noise 5%
 - Mosaic
- Chia bộ dữ liệu thành các tập
 - Train 82% là 723 ảnh
 - Validate 14% là 120 ảnh.
 - Test 4% là 39 ảnh

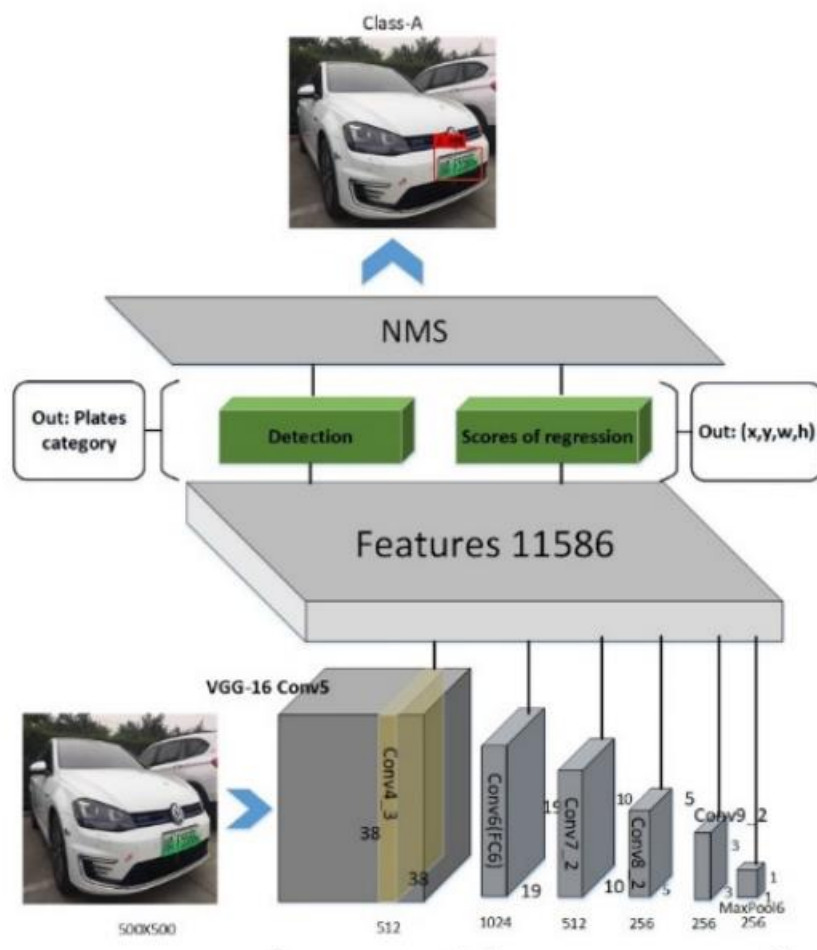
Một số hình ảnh trong tập dữ liệu sau khi được tăng cường:



III. CÁC NGHIÊN CỨU LIÊN QUAN

1. Phương pháp nhận dạng biển số xe dựa trên kiến trúc mạng LPR-SSD

Lớp trích xuất đặc trưng nhận dạng biển số xe bao gồm 5 lớp tích chập và 1 lớp gộp tối đa. Bản đồ đặc trưng cho mỗi đầu ra lớp tích chập được sử dụng cho offset của hộp mặc định và dự đoán điểm số của các loại biển số khác nhau. Trên các bản đồ đặc trưng này, việc huấn luyện và dự đoán vị trí biển số và phân loại được thực hiện để đạt được phát hiện nhiều tỷ lệ. Sau khi trích xuất đặc trưng của biển số, vị trí biển số và loại biển số được xuất ra. Cuối cùng, thông qua sàng lọc Non-Maximum Suppression (NMS), kết quả định vị và phân loại cuối cùng được xuất ra



2. Nhận dạng biển số xe tự động dựa trên artificial neural networks (ANN)

Ưu điểm của hệ thống này là có khả năng nhận dạng biển số xe một cách tự động và hiệu quả với tỷ lệ nhận dạng đúng là 95,36% Tối đa 5000 lần lặp lại được thực hiện cho mỗi tập hợp đầu vào. Khi hệ thống đạt đến tỷ lệ lỗi tối thiểu được xác định bởi người dùng, các lần lặp lại sẽ bị dừng lại. Tỷ lệ lỗi tối thiểu được xác

định cho ứng dụng này là 0,001. Chỉ một trong các hình ảnh đầu vào được sử dụng để kiểm tra hệ thống, phần còn lại được thực hiện trong giai đoạn đào tạo. Hệ thống này bao gồm ba chủ đề chính. Đó là xác định vùng biển số từ hình ảnh xe, phân đoạn các ký tự từ hình ảnh biển số và nhận dạng các ký tự đã phân đoạn



3. Nhận dạng biển số xe tự động dựa trên Convolutional Neural Network (CNN)

Nhận dạng ký tự quang học (OCR) là một công nghệ chủ yếu được sử dụng để nhận dạng văn bản in máy hoặc viết tay của con người trong các tài liệu quét, hình ảnh và sau đó chuyển đổi thành dạng có thể chỉnh sửa. Mở rộng ứng dụng của nó, chúng ta có thể sử dụng OCR trong hệ thống giao thông được hướng dẫn bởi máy tính, tức là một hệ thống giao thông thông minh có thể hoạt động một mình với ít hoặc không có sự can thiệp của con người. Nhận dạng biển số xe sẽ đóng vai trò quan trọng trong việc xây dựng bất kỳ hệ thống giao thông thông minh nào. Sau khi lấy biển số, hệ thống sẽ phân đoạn ký tự trên biển số bằng cách quét ngang hình ảnh. Sau đó, với sự trợ giúp của CNN, chúng tôi nhận ra các ký tự. Sau khi nhận dạng các ký tự, chúng ta có thể sử dụng chúng để xác thực biển số hoặc trích xuất thông tin về chủ sở hữu xe.

Lưu đồ của hệ thống được đề xuất được hiển thị trong hình bao gồm các bước chính sau:

Bước 1: Đã nhận được hình ảnh đầu vào

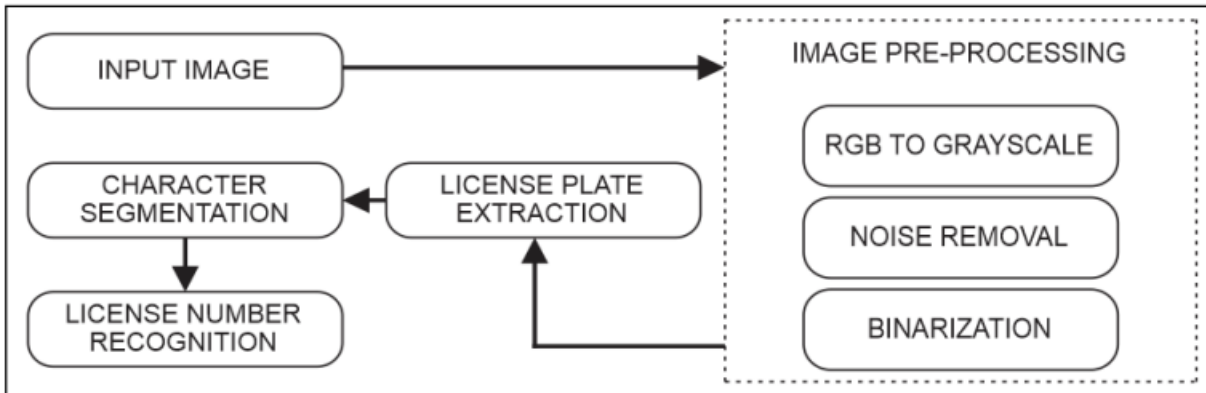
Bước 2: Tiền xử lý ảnh

- Chuyển đổi RGB sang thang độ xám.
- Loại bỏ tiếng ồn bằng Lọc song phương lặp lại
- Nhị phân hóa hình ảnh

Bước 3: Trích xuất biển số bằng thuật toán phát hiện cạnh/ thuật toán biên của Sobel

Bước 4: Phân đoạn nhân vật

Bước 5: Nhận dạng số Giấy phép sử dụng mạng thần kinh tích chập



IV. YOLO V5

1. Giới thiệu

YOLOV5 là một mô hình phát hiện đối tượng nhanh và hiệu quả được phát triển bởi Ultralytics. Nó là phiên bản nâng cấp của YoloV4 với các cải tiến về tốc độ, độ chính xác và khả năng tùy chỉnh.

Một số tính năng:

- Tăng tốc phát hiện đối tượng: Với kiến trúc đơn giản hóa và sử dụng các kỹ thuật tối ưu mới, YOLOV5 có thể phát hiện đối tượng nhanh hơn so với phiên bản trước đó.
- Khả năng tùy chỉnh linh hoạt: YOLOV5 cho phép người dùng tùy chỉnh mô hình để phù hợp với các nhu cầu riêng của họ, bao gồm kích thước đầu vào, kiến trúc mạng, số lượng lớp đối tượng, v.v.
- Độ chính xác cao: YOLOV5 đạt được kết quả tốt trong các bài kiểm tra thực tế về phát hiện đối tượng, với độ chính xác và độ nhạy cao.
- Dễ sử dụng: YOLOV5 được viết bằng Python và có các công cụ hỗ trợ sử dụng, cho phép người dùng dễ dàng triển khai và sử dụng nó.

2. Train model

Bước 1: Cài đặt YOLOv5 và các dependencies

```

!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!git reset --hard 064365d8683fd002e9ad789c1e91fa3d021b44f0
  
```

Bước 2: Export dataset từ Roboflow theo format YOLO v5 PyTorch

Export

Format

YOLO v5 PyTorch

TXT annotations and YAML config used with [YOLOv5](#).

☐ download zip to computer
 ☒ show download code

Cancel

Continue

Your Download Code

Jupyter

Terminal

Raw URL

Paste this snippet into [a notebook from our model library >>](#) to download and unzip [your dataset >>](#):

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="")
project = rf.workspace("team11").project("license-plates-ft9my")
dataset = project.version(3).download("yolov5")
```

Bước 3: Xác định cấu hình và kiến trúc cho mô hình YOLOv5 (không bắt buộc)

Ta sẽ viết một tập lệnh yaml xác định các tham số cho mô hình như number of classes, anchors, and each layer.

```
# define number of classes based on YAML
import yaml
with open(dataset.location + "/data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

Bước 4: Thiết lập trọng số và train model

```
!python train.py --img 704 --batch 16 --epochs 999 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results --cache
```

Trong đó:

- img 704: 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 999: Số lượng epochs (vòng lặp huấn luyện) để huấn luyện mô hình.
- weights yolov5s.pt: Đường dẫn đến trọng số được sử dụng để khởi tạo mô hình huấn luyện. Trong trường hợp này, nhóm em sử dụng trọng số của YoLov5s.

Lý do: YoLov5s là một phiên bản nhỏ gọn và hiệu quả của YoLov5, phù hợp cho các ứng dụng thời gian thực hoặc các thiết bị có tài nguyên hạn chế, nhưng vẫn đảm bảo độ chính xác cao.

Model	size (pixels)	mAP ^{val} 50-95	mAP ^{val} 50	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

V. OPTICAL CHARACTER RECOGNITION

1. Giới thiệu

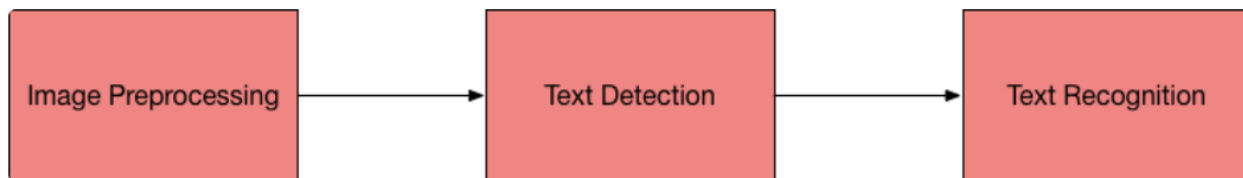
OCR là thuật ngữ được viết tắt bởi cụm từ Optical Character Recognition (dịch là: nhận dạng ký tự quang học). Đây là ứng dụng công nghệ chuyên dùng để đọc text ở file ảnh. Được biết đến là một công cụ scan kỹ thuật số chuyên nhận dạng các ký tự, chữ viết tay, hay chữ đánh máy, công nghệ này chuyên dùng để truyền tải, nhập liệu dữ liệu. Đặc biệt, ở OCR có khả năng kỹ thuật số nhiều dưới nhiều dạng tài liệu khác nhau: hóa đơn, hộ chiếu, danh thiếp, tài liệu...

Đến với OCR, những văn bản số hóa, tìm kiếm và chỉnh sửa sẽ được thực hiện điện tử. Đồng thời, chúng giúp tiết kiệm không gian lưu trữ tài liệu bằng việc hiển thị trên trực tiếp.

Khi một trang in hoặc viết tay được quét, nó được lưu dưới dạng tệp ảnh xạ bit có định dạng TIF. Chúng ta có thể đọc hình ảnh này khi nó được hiển thị trên màn hình. Tuy nhiên, với máy tính, nó chỉ là một loạt các dấu chấm màu trắng và đen. Nó nhìn vào từng dòng của hình ảnh và xác định xem chuỗi các chấm có khớp với một số hoặc chữ cái cụ thể không. Kể từ khi nhận dạng văn bản bằng máy học gia tăng vào năm 2014, mọi người đã thử một số kỹ thuật truyền thống để giải quyết vấn đề Máy học OCR thông qua thị giác máy tính. Tuy nhiên, điều cần thiết là khám phá các phương pháp mới để làm cho các mô hình của chúng tôi trở nên mạnh mẽ trước những biến thể này để các doanh nghiệp có thể triển khai các ứng dụng máy học của họ trên quy mô lớn.

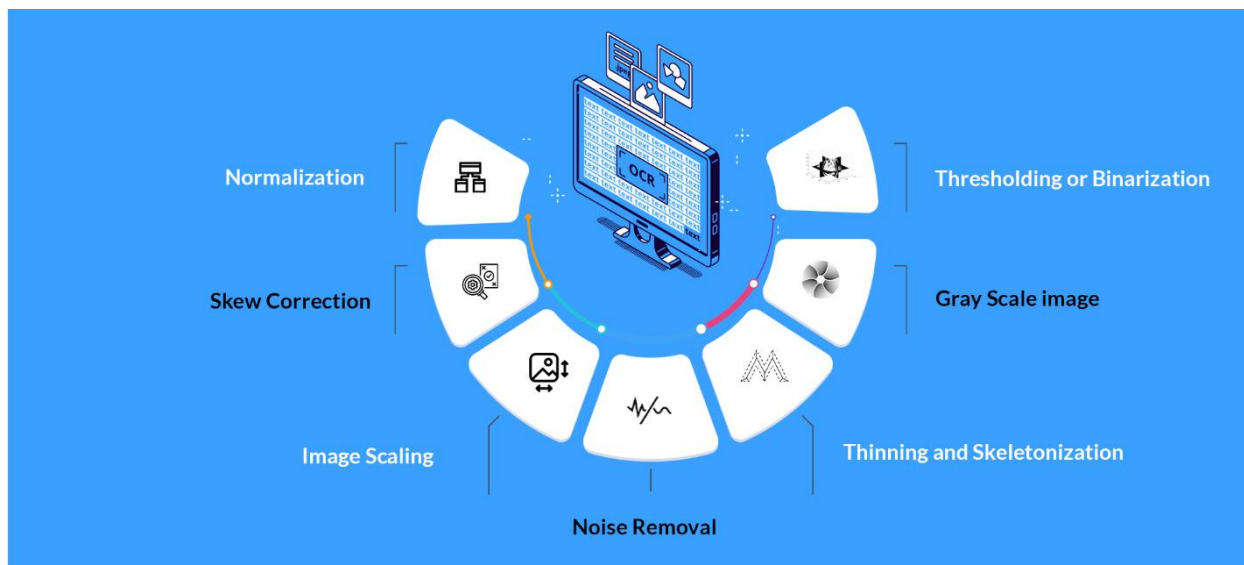
2. Các bước thực hiện bài toán OCR

Bất kỳ quy trình OCR máy học điển hình nào cũng tuân theo các bước sau:



2.1. Preprocessing

Chúng ta có 7 bước thực hiện:



a. Normalization

Quá trình này thay đổi phạm vi giá trị cường độ điểm ảnh. Mục đích của việc thực hiện chuẩn hóa là đưa hình ảnh về phạm vi bình thường để cảm nhận. OpenCV sử dụng hàm `normalize()` để chuẩn hóa hình ảnh.

b. Skew Correction

Trong khi quét hoặc chụp ảnh bất kỳ tài liệu nào, đôi khi hình ảnh được quét hoặc chụp có thể bị lệch một chút. Để OCR có hiệu suất tốt hơn nên xác định độ lệch của hình ảnh và sửa nó.

c. Image Scaling

Để đạt được hiệu suất OCR tốt hơn, hình ảnh phải có hơn 300 PPI (pixel trên inch). Vì vậy, nếu kích thước hình ảnh dưới 300 PPI, chúng ta cần tăng nó lên.

d. Noise Removal

Bước này loại bỏ các chấm/mảng nhỏ có cường độ cao so với phần còn lại của hình ảnh để làm mịn hình ảnh. Chức năng `NI Means Denoising Colored` nhanh của OpenCV có thể thực hiện điều đó một cách dễ dàng. **Thinning and Skeletonization**

Bước này được thực hiện cho văn bản viết tay, vì những người viết khác nhau sử dụng các độ rộng nét khác nhau để viết. Bước này làm cho độ rộng của các nét đều nhau. Điều này có thể được thực hiện trong OpenCV

e. Gray Scale image

Quá trình này chuyển đổi một hình ảnh từ các không gian màu khác sang các sắc thái của Xám. Màu sắc thay đổi giữa đen hoàn toàn và trắng hoàn toàn. Chức năng `cvtColor()` của OpenCV thực hiện nhiệm vụ này rất dễ dàng.

f. Thresholding or Binarization

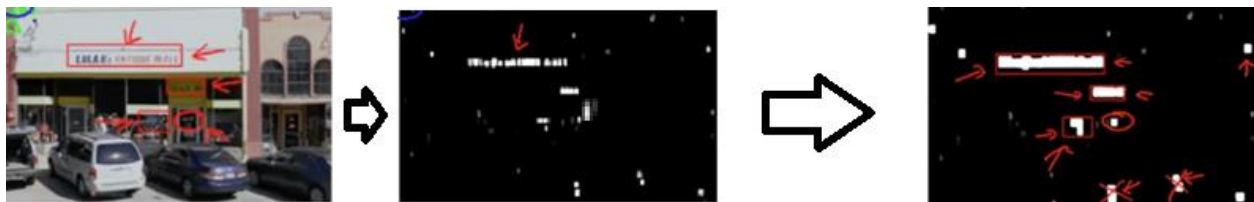
Bước này chuyển đổi bất kỳ hình ảnh màu nào thành hình ảnh nhị phân chỉ chứa hai màu đen và trắng. Nó được thực hiện bằng cách cố định một ngưỡng (thường là một nửa phạm vi pixel 0-255, tức là 127). Giá trị pixel lớn hơn ngưỡng được chuyển đổi thành pixel trắng khác thành pixel đen. Để xác định giá trị ngưỡng theo hình ảnh Binarization và Adaptive Binarization của Otsu có thể là lựa chọn tốt hơn. Trong OpenCV, điều này có thể được thực hiện như đã cho.

OCR có nhiều lĩnh vực ứng dụng trong thế giới thực và việc cải thiện hiệu suất của các mô hình OCR là cần thiết để tránh những sai lầm trong thế giới thực. Quá trình xử lý trước hình ảnh giúp giảm lỗi ở một mức đáng kể và giúp thực hiện OCR tốt hơn. Các bước tiền xử lý hình ảnh có thể được quyết định dựa trên hình ảnh có sẵn để trích xuất văn bản. Dựa trên hình ảnh, một số bước có thể được loại bỏ và một số bước khác có thể được thêm vào theo yêu cầu. Quá trình tiền xử lý trở nên hiệu quả hơn khi được áp dụng sau khi hiểu rõ hơn về dữ liệu đầu vào (hình ảnh) và tác vụ cần thực hiện.

2.2. Text Detection

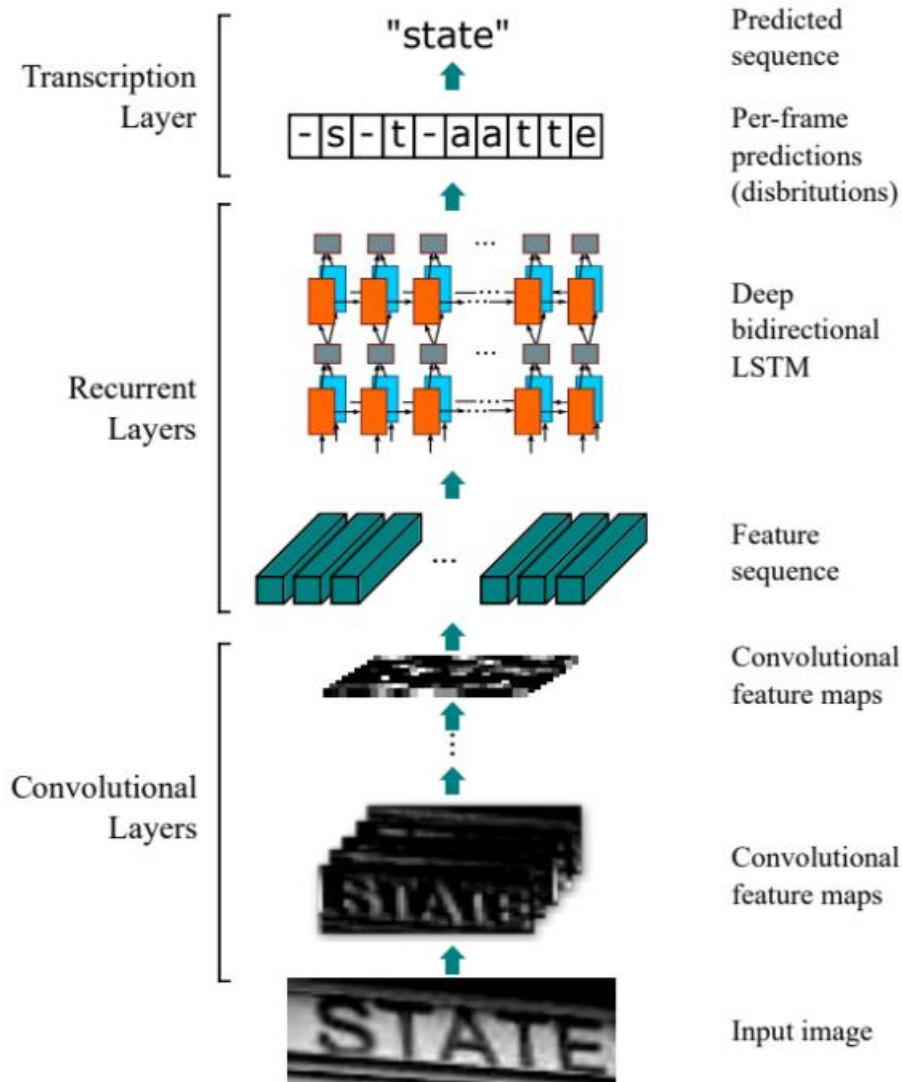
Các bước thực hiện

- Nếu ($y = 1$), có văn bản
- Nếu ($y = 0$), không có văn bản
- chạy một trình phân loại cửa sổ trượt trên hình ảnh
 - Chúng tôi có (ở phía dưới bên trái) các vùng màu trắng biểu thị các vùng văn bản
 - Trắng sáng: trình phân loại xuất ra xác suất văn bản ở vị trí rất cao
- Nếu chúng ta lấy thêm một văn bản bằng cách lấy đầu ra của bộ phân loại và áp dụng toán tử mở rộng
 - Nó lấy vùng trắng và mở rộng chúng
 - Nếu chúng ta sử dụng phương pháp phỏng đoán và loại bỏ những phương pháp có tỷ lệ chiều cao trên chiều rộng bất thường



2.3. Text Recognition

Convolutional Recurrent Neural Network (CRNN) là sự kết hợp của CNN, RNN và CTC (Phân loại tạm thời của nhà kết nối) đối với các tác vụ nhận dạng trình tự dựa trên hình ảnh, chẳng hạn như nhận dạng văn bản cảnh và OCR. Kiến trúc mạng đã được lấy từ bài báo này được xuất bản vào năm 2015.



Kiến trúc mạng này tích hợp tính năng trích xuất, mô hình hóa trình tự và sao chép vào một khung thống nhất. Mô hình này không cần phân đoạn ký tự. Mạng thần tích chập trích xuất các tính năng từ hình ảnh đầu vào (vùng phát hiện văn bản). Mạng thần kinh tái phát hai chiều sâu dự đoán chuỗi nhãn với một số mối quan hệ giữa các ký tự. Lớp phiên mã chuyển đổi mỗi khung do RNN tạo thành một chuỗi nhãn.

3. Ưu điểm và Nhược điểm

3.1. Ưu Điểm

Phát hiện nhiều ngôn ngữ với độ chính xác cao: Với AI OCR tiên tiến hiện tại, các chương trình OCR có thể phát hiện và hiểu các ký tự phức tạp. Nó cũng có thể học cách ghép các từ giữa một số ngôn ngữ, giúp nâng cao hơn nữa khả năng dịch thuật. Để cho ví dụ thì nổi bật nhất là Tesseract, hệ thống OCR do Google phát triển, phát hiện văn bản bằng 100 ngôn ngữ, bao gồm cả các ngôn ngữ viết từ phải sang trái như tiếng Ả Rập và tiếng Do Thái.

Xác định văn bản không có cấu trúc: Giống như đã trình bày ở trên, hệ thống OCR tích hợp học sâu (deep learning) sẽ xây dựng một mạng lưới nơron mô phỏng lại não người, giúp hệ thống không chỉ nhận dạng văn bản mà còn xác định được những trường văn bản có nghĩa.

Nhận biết các điều kiện thay đổi: Sử dụng một loạt công nghệ bao gồm học máy, xử lý ngôn ngữ tự nhiên và xử lý hình ảnh, công nghệ OCR có thể học các ngôn ngữ mới và điều chỉnh để thay đổi các loại tài liệu. Machine Learning cho phép AI OCR nhận diện chữ trên các loại văn bản khác nhau một cách nhanh chóng với độ chính xác vượt trội, không cần nhóm phát triển của bạn phải điều chỉnh thủ công

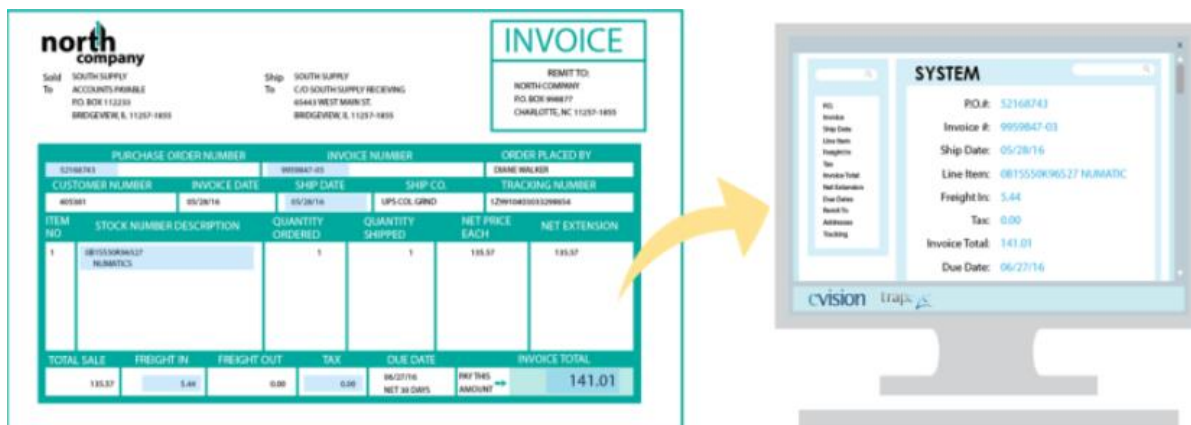
Mở rộng khả năng của phần mềm nhận diện chữ: Công nghệ OCR truyền thống chỉ dừng lại ở việc giải quyết bài toán số hóa văn bản. Đối với AI OCR, ngoài nhận diện ký tự, phần mềm có thể phân tích dữ liệu đó, biến nó thành thông tin có ích với doanh nghiệp. Ví dụ, một ngân hàng sử dụng OCR để quét dữ liệu khách hàng vay vốn và lưu thành hồ sơ trên hệ thống. Nếu AI OCR có thể phân tích các thông tin đó và trả về kết quả là mức độ rủi ro của chúng thì giá trị mang lại sẽ nhiều hơn.

3.2. Nhược Điểm

Nhược điểm của OCR là không có khả năng đọc chữ viết tay và các phong chữ quá phức tạp.

4. Ứng dụng của OCR

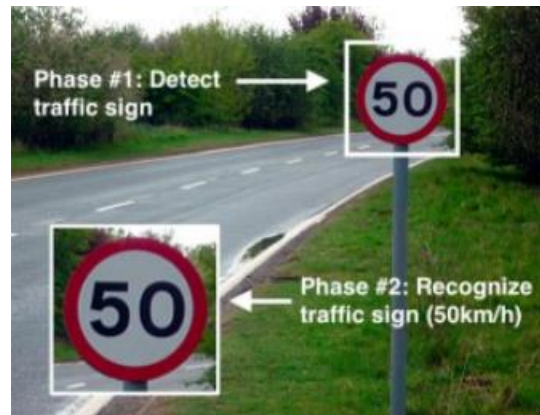
Automatic Data Entry - Tự động nhập liệu: Đây có lẽ là ứng dụng phổ biến và quan trọng nhất của OCR. Trước đây, đối với những số liệu trong một hình ảnh hay tài liệu scan, để đưa vào máy tính xử lý, con người phải nhập thủ công bằng tay. Việc này rất mất thời gian và nhàm chán. Ngày nay, với sự hỗ trợ của OCR, quá trình này diễn ra hoàn toàn tự động, nhanh chóng, dễ dàng, độ chính xác cao.



Nhận diện biển số xe: Áp dụng trong các bãi đỗ xe, tự động nhận diện biển số giúp giảm thời gian quản lý cho cả người lái xe và nhân công bảo vệ.



Xe tự lái: OCR giúp xe tự động nhận diện biển số để đi theo đúng chỉ dẫn



VI. ĐÁNH GIÁ MÔ HÌNH

1. Mô hình Yolo v5

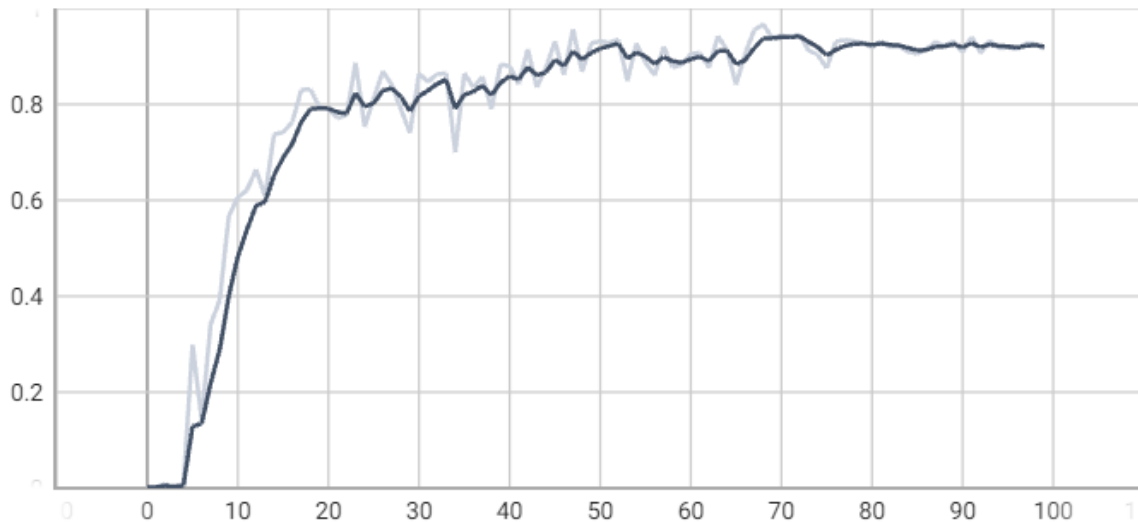
Mô hình được huấn luyện qua 344 epochs và được đánh giá qua độ đo precision, recall và mean average precision.

a. Precision

Precision đo lường tỷ lệ giữa số lượng các bounding box phát hiện đúng (true positives) và tổng số bounding box được dự đoán là đúng (true positives + false positives). Chỉ số P cho biết khả năng của mô hình để tránh phát hiện sai các bounding box.

Precision trong mô hình Yolo v5 kết quả 0.966, tức là mô hình có khả năng dự đoán chính xác khoảng 96.6% các bounding box phát hiện được.

metrics/precision

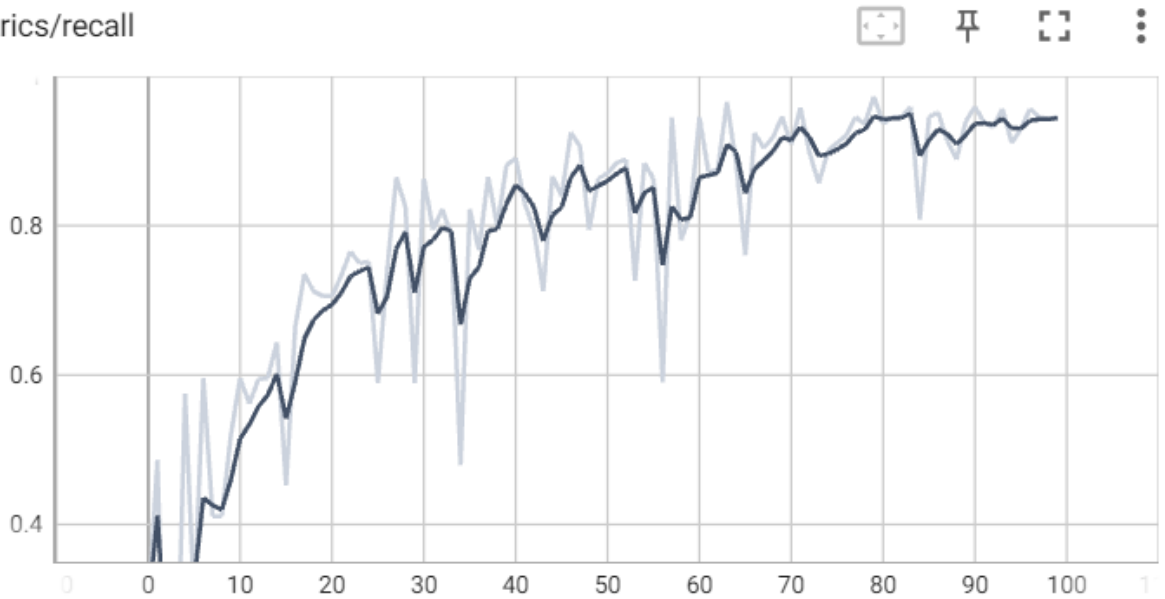


b. Recall

Recall tính toán tỷ lệ giữa số lượng các bounding box phát hiện đúng (true positives) và tổng số bounding box thực tế (true positives + false negatives). Chỉ số R đo lường khả năng của mô hình để tìm ra tất cả các bounding box chứa đối tượng.

Recall trong mô hình có kết quả là 0.979, cho thấy mô hình có khả năng tìm ra khoảng 97.9% các bounding box chứa đối tượng trong tập dữ liệu.

metrics/recall

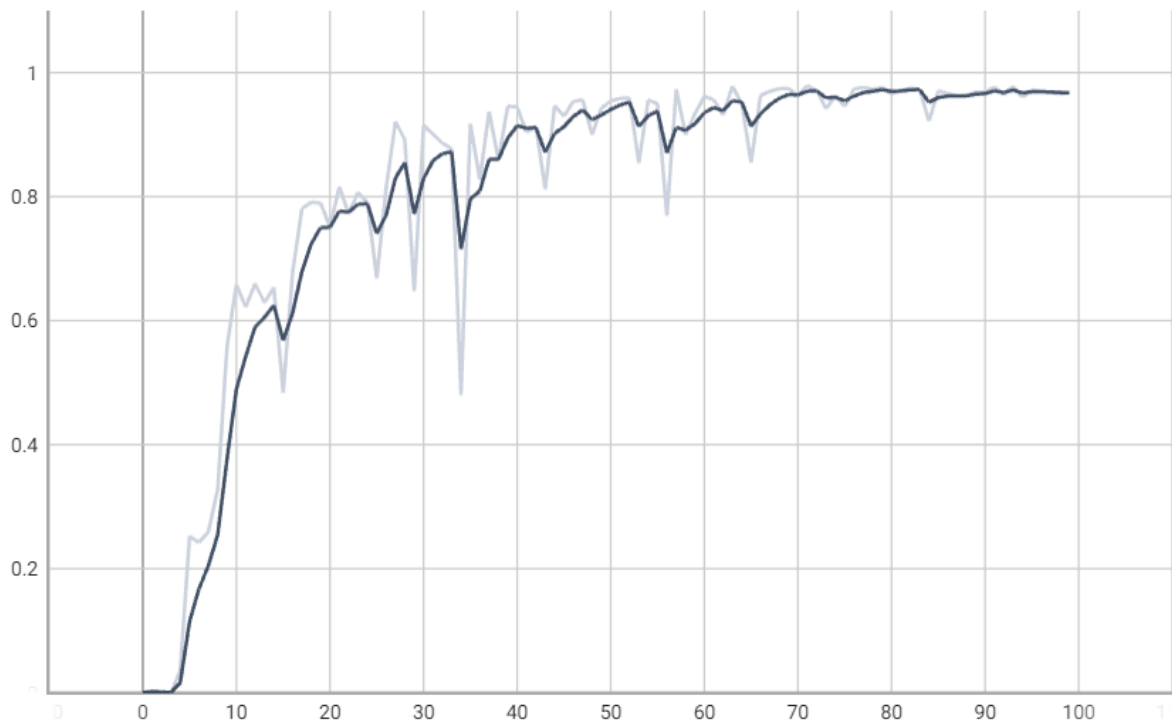


c. mAP (mean Average Precision)

mAP là một phép tổng hợp của các giá trị AP (Average Precision) trên tất cả các lớp đối tượng. AP được tính bằng cách tính toán diện tích dưới đường cong Precision-Recall cho mỗi lớp. Sau đó, mAP là trung bình của các giá trị AP này. mAP là một chỉ số quan trọng để đánh giá hiệu suất tổng thể của mô hình YOLOv5 trong việc phát hiện và phân loại đối tượng.

mAP đạt kết quả 0.992, chỉ số này đo lường khả năng của mô hình trong việc phát hiện và phân loại các đối tượng với ngưỡng xác suất là 0.5. Điều này cho thấy mô hình có hiệu suất rất cao trong việc phát hiện đối tượng.

metrics/mAP_0.5



2. Mô hình OCR

Mô hình OCR (Optical Character Recognition) có thể được đánh giá dựa trên hai yếu tố chính: thời gian và độ chính xác. Thời gian chạy của mô hình bị ảnh hưởng bởi kích thước của ảnh đầu vào. Điều này có nghĩa là mô hình sẽ mất nhiều thời gian hơn để xử lý các ảnh lớn hơn. Để đánh giá thời gian, chúng ta có thể thực hiện đo thời gian xử lý cho mỗi ảnh với các kích thước khác nhau.

Độ chính xác của mô hình OCR phụ thuộc vào độ nét của ảnh. Khi ảnh không rõ nét, mô hình có thể gặp khó khăn trong việc nhận dạng chính xác các ký tự. Để đánh giá độ chính xác, chúng ta có thể sử dụng tập dữ liệu kiểm tra và tính toán tỷ lệ ký tự nhận dạng chính xác trên tổng số ký tự trong tập dữ liệu. Đồng thời, có thể điều chỉnh độ nét của ảnh và xem xét sự thay đổi trong độ chính xác để hiểu sự tương quan giữa độ nét và độ chính xác.

Tuy nhiên, việc đánh giá mô hình OCR cần phải dựa trên tập dữ liệu đánh giá đủ lớn và đa dạng để đảm bảo tính chính xác và độ tin cậy của kết quả đánh giá. Bên cạnh đó, việc tinh chỉnh các thông số và siêu tham số của mô hình, cũng như sử dụng các kỹ thuật tiền xử lý ảnh phù hợp, có thể cải thiện thời gian và độ chính xác của mô hình OCR.

VII. DEPLOY MODEL

1. Giới thiệu

Flask:

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

Docker:

Docker là nền tảng cung cấp cho các công cụ, service để các developers, admins/systems có thể phát triển, thực thi, chạy các ứng dụng với containers. Hay nói một cách khác nó là một nền tảng để cung cấp cách để building, deploy và run các ứng dụng một cách dễ dàng trên nền tảng ảo hóa - "Build once, run anywhere". Hay nói một cách dễ hiểu như sau: Khi chúng ta muốn chạy app thì chúng ta phải thiết lập môi trường chạy cho nó. Thay vì chúng ta sẽ đi cài môi trường chạy cho nó thì chúng ta sẽ chạy docker.

Ứng dụng Docker chạy trong vùng chứa (container) có thể được sử dụng trên bất kỳ hệ thống nào: máy tính xách tay của nhà phát triển, hệ thống trên cơ sở hoặc trong hệ thống đám mây. Và là một công cụ tạo môi trường được "đóng gói" (còn gọi là Container) trên máy tính mà không làm tác động tới môi trường hiện tại của máy, môi trường trong Docker sẽ chạy độc lập.

Docker có thể làm việc trên nhiều nền tảng như Linux, Microsoft Windows và Apple OS X.

Trong báo cáo này chúng em có build image Docker để tiện cho việc cài đặt môi trường và chạy.

```
docker pull nguyenvhuc0804/myapp
```

2. Workflow

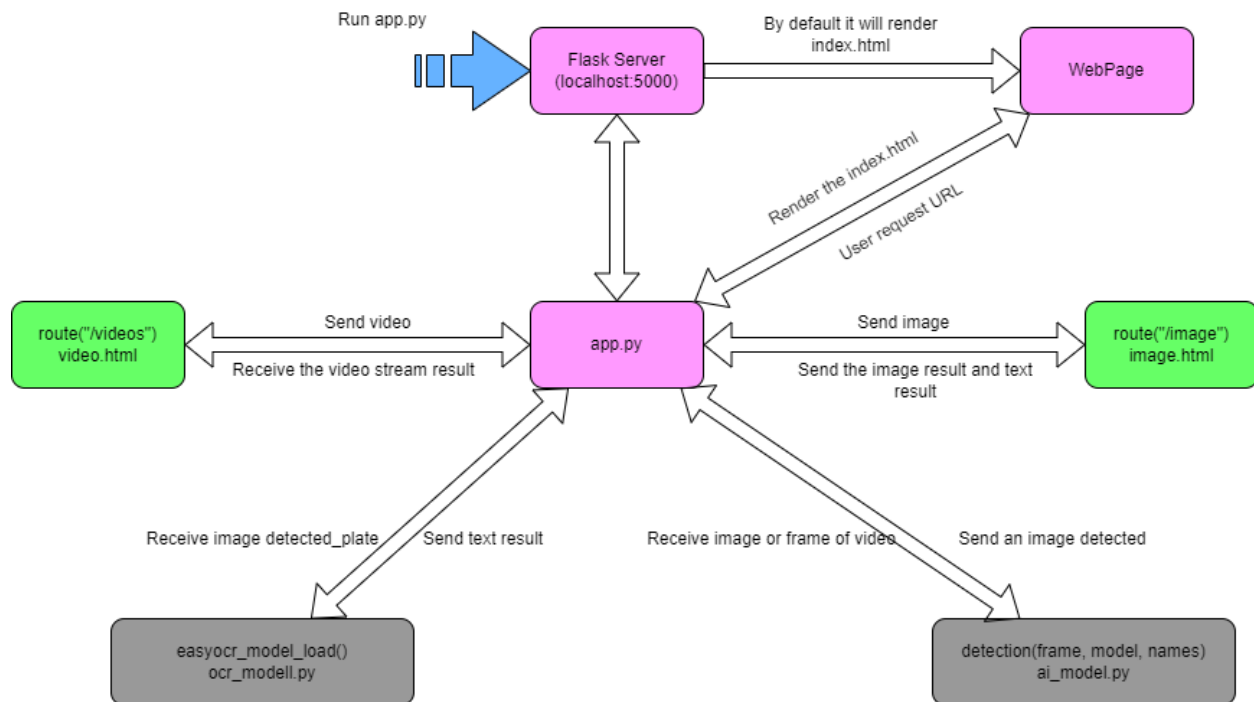
Người dùng truy cập vào ứng dụng web thông qua trình duyệt và được chuyển tới giao diện chính của trang web, index.html, thông qua Flask Server đang chạy trên cổng 5000.

Giao diện chính của trang web hiển thị hai tùy chọn chính cho người dùng:

- Router("/videos"): Cho phép người dùng chọn một video để phát hiện biển số xe trong video. Khi người dùng chọn video, các khung hình (frame) trong video sẽ được gửi từng cái một tới ứng dụng app.py để phân tích. Các khung hình này được gửi qua AI_model.py để phát hiện biển số xe và OCR_model.py để xử lý đọc biển số. Sau đó, các khung hình đã được phân tích và biển số xe đã được detect sẽ được trả về và render lần lượt trên màn hình web.
- Router("/image"): Cho phép người dùng chọn một ảnh để phát hiện biển số xe trong ảnh. Khi người dùng chọn ảnh, ảnh này sẽ được gửi tới ứng dụng app.py. Tương tự như trên, ảnh được gửi qua AI_model.py để phát hiện biển số xe và OCR_model.py để xử lý đọc biển số. Sau đó, ảnh đã được phân tích và biển số xe đã được detect sẽ được trả về và render trên giao diện web cùng với văn bản của biển số xe.

Trong quá trình thực hiện các chức năng trên, app.py đóng vai trò quản lý và điều phối ứng dụng. Nó tương tác với AI_model.py để phân tích ảnh và AI_model.py để xử lý đọc biển số. App.py nhận các frame ảnh từ router("/videos") hoặc ảnh từ router("/image"), sau đó gửi chúng qua AI_model.py và OCR_model.py để thực hiện các tác vụ phát hiện biển số và xử lý đọc biển số. Kết quả từ cả hai mô hình này sau đó được trả về app.py để render lên giao diện web.

Tóm lại, ứng dụng sử dụng Flask Server để chạy trên cổng 5000 và có hai router chính: "/videos" để chọn video và phát hiện biển số trong video, và "/image" để chọn ảnh và phát hiện biển số trong ảnh. App.py quản lý và điều phối quá trình xử lý, tương tác với AI_model.py để phát hiện biển số và OCR_model.py để xử lý đọc biển số. Kết quả được trả về và render trên giao diện web cho người dùng xem.



Giao diện của ứng dụng web trong đồ án của nhóm

Number Plate Recognition Image Video

CS338 - NHẬN DẠNG

Nhận dạng biển số xe sử dụng Yolo v5 và OCR

Giảng viên: Đỗ Văn Tiến

Sinh viên thực hiện	Mã số sinh viên
Nguyễn Thanh Phúc	20521769
Nguyễn Minh Nhật	20521708
Lê Thế Tuấn	20522113
Trần Minh Phúc	20521782

Choose image file



Choose file

No file chosen

Upload

Detected Image



Detected Text

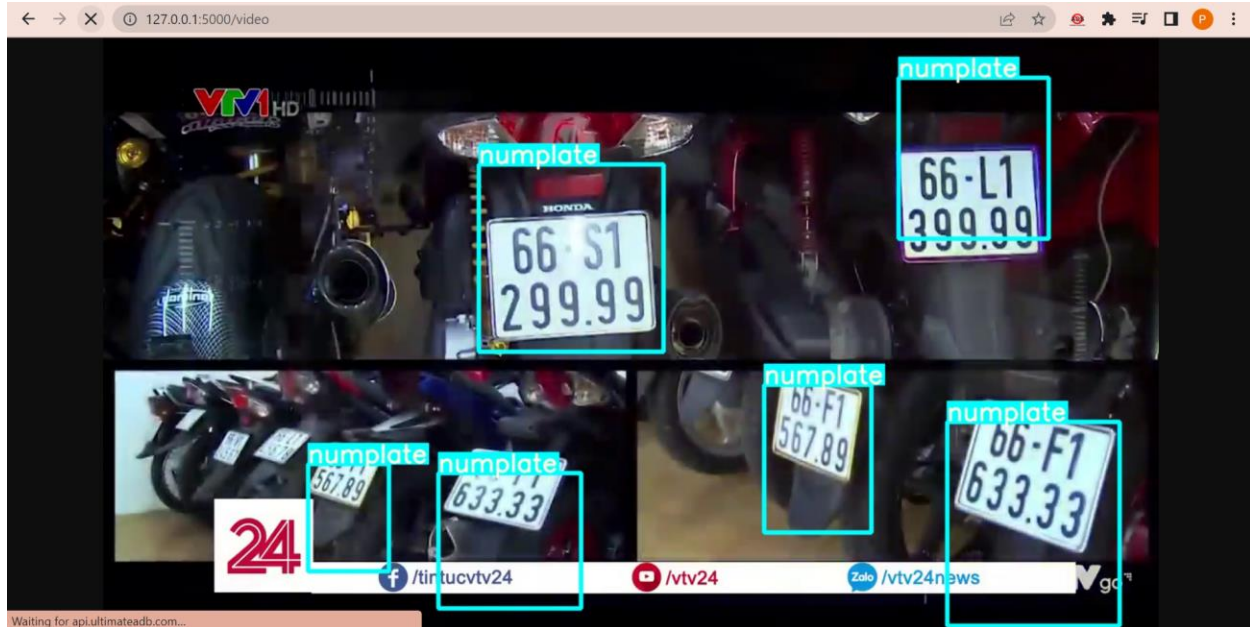
- numplat
- 51F.888.38

Video Object
Detection

Choose file

No file chosen

Detect



VIII. KẾT LUẬN

Trong đồ án này, chúng em đã triển khai một hệ thống nhận dạng biển số xe sử dụng YOLOv5 và OCR (Optical Character Recognition). Kết quả đạt được là một hệ thống hiệu quả và chính xác có khả năng nhận dạng biển số xe từ các hình ảnh và video.

Đầu tiên, chúng em đã sử dụng YOLOv5, một mô hình deep learning, để phát hiện và xác định vị trí của các biển số xe trong ảnh hoặc video. YOLOv5 cho phép chúng em tìm thấy các vùng quan tâm nhanh chóng và chính xác.

Tiếp theo, chúng em đã áp dụng OCR để trích xuất các ký tự từ vùng biển số đã được xác định. OCR sử dụng các thuật toán và mô hình nhận dạng ký tự để chuyển đổi hình ảnh chứa văn bản thành văn bản có thể đọc được. Qua đó, chúng em đã thu được dãy ký tự tương ứng với biển số xe.

Hệ thống nhận dạng biển số xe của chúng em đã được đánh giá và đạt được kết quả tốt trên tập dữ liệu kiểm tra. Độ chính xác của hệ thống được đo bằng các chỉ số như độ chính xác, độ phủ và mAP.

IX. HƯỚNG PHÁT TRIỂN

Mặc dù đồ án đã đạt được kết quả khả quan, nhưng còn một số hướng phát triển tiềm năng để nâng cao hiệu suất và khả năng ứng dụng của hệ thống nhận dạng biển số xe.

- Mở rộng tập dữ liệu: Để tăng tính đa dạng và độ phủ của hệ thống, chúng em có thể thu thập thêm dữ liệu từ nhiều nguồn khác nhau, bao gồm các biển số xe trong điều kiện ánh sáng khác nhau, góc chụp khác nhau và các loại biển số xe khác nhau. Điều này sẽ giúp cải thiện khả năng tổng quát hóa của mô hình.

- Tối ưu hóa mô hình YOLOv5: Chúng em có thể tiến hành tối ưu hóa mô hình YOLOv5 bằng cách tăng kích thước của mạng, sử dụng các kỹ thuật data augmentation, hoặc thử các phiên bản YOLOv5 khác nhau như YOLOv5x để đạt được độ chính xác cao hơn.
- Cải thiện OCR: Chúng em sẽ tiếp tục nghiên cứu và cải thiện OCR để xử lý các trường hợp khó khăn như biển số xe có ký tự nhỏ, mờ, hoặc bị che khuất. Chúng em có thể thử nghiệm các mô hình OCR tiên tiến hoặc kết hợp với các kỹ thuật xử lý ảnh khác để tăng cường khả năng nhận dạng.
- Tích hợp vào ứng dụng thực tế: Chúng em sẽ phát triển một giao diện người dùng đơn giản và thân thiện để ứng dụng hệ thống nhận dạng biển số xe vào các ứng dụng thực tế, chẳng hạn như hệ thống giám sát giao thông hoặc bãi đỗ xe tự động.
- Đánh giá và tinh chỉnh: Chúng em sẽ tiến hành đánh giá và tinh chỉnh thêm hệ thống nhận dạng biển số xe bằng cách thu thập phản hồi từ người dùng thực tế và thực hiện các bước cải tiến dựa trên phản hồi đó.
- Với những hướng phát triển này, hệ thống nhận dạng biển số xe có thể trở nên mạnh mẽ và ứng dụng rộng rãi trong các lĩnh vực như an ninh giao thông, quản lý đô thị và phân tích dữ liệu xe cộ.

REFERENCE

- Jacob Solawetz, J. N. (n.d.). *Roboflow*. Retrieved from Roboflow:
<https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/>
- JaiedTeam. (n.d.). *EasyOCR*. Retrieved from EasyOCR:
<https://github.com/JaiedAI/EasyOCR>
- Jocher, G. (n.d.). *yolov5*. Retrieved from yolov5: <https://github.com/ultralytics/yolov5>
- PyImageSearch. (n.d.). *Training the YOLOv5*. Retrieved from Training the YOLOv5:
<https://pyimagesearch.com/2022/06/20/training-the-yolov5-object-detector-on-a-custom-dataset/>
- Thủy, N. T. (n.d.). *Viblo*. Retrieved from Viblo: <https://viblo.asia/p/su-dung-colab-train-yolov5-voi-custom-dataset-phat-hien-cac-doi-tuong-dac-thu-Az45bqv6lxY>