

Ngày công bố Tháng Ba 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Nhận dạng biển số

Hồ Thiên Phước¹, Thành viên, IEEE, Lưu Thị Hồng Ngọc², Thành viên, IEEE

¹Đại học khoa học tự nhiên, HCM, 70000 VN

Giáo viên hướng dẫn: Thầy Võ Hoài Việt phụ trách bộ môn Thị giác máy tính

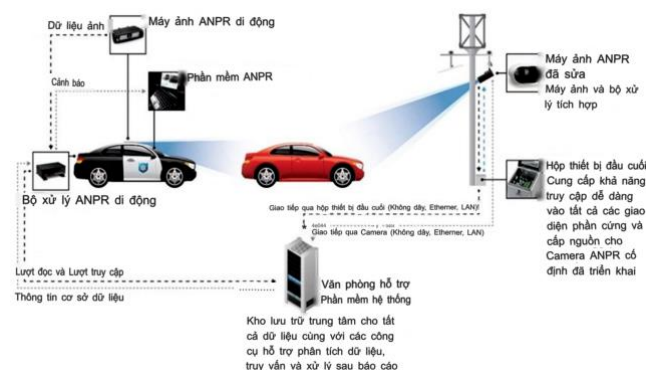
Sinh viên thực hiện: Một, Hồ Thiên Phước (e-mail: 19127517@student.hcmus.edu.vn). Hai, Lưu Thị Hồng Ngọc (email: lthngoc19@clc.fitus.edu.vn).

Tài liệu thông tin về công trình liên quan từ công nghệ phát hiện, nhận dạng biển số; Các tầng cơ bản của hệ thống thị giác; Tài liệu tham khảo.

ABSTRACT Công nghệ ANPR có khả năng phát hiện và nhận dạng phương tiện đi đường thông qua biển số xe. Tình trạng biển số, định dạng không chuẩn, cảnh nền phức tạp, chất lượng máy ảnh, vị trí lắp camera, khả năng chịu biến dạng, chuyển động mờ, vấn đề tương phản, chỉnh sửa lại, xử lý và giới hạn bộ nhớ, điều kiện môi trường, ảnh chụp trong nhà / ngoài trời hoặc ngày/ đêm, phần mềm- các công cụ khác hoặc ràng buộc dựa trên phần cứng có thể làm giảm hiệu suất của quá trình nhận dạng. Các kỹ thuật Deep Learning được sử dụng rộng rãi trong lĩnh vực Thị giác máy tính để có tỷ lệ phát hiện tốt hơn.

I. GIỚI THIỆU

Với sự ra đời của điện thoại thông minh, nhiều ANPR cũng đã trở thành thiết bị cầm tay. Do chi phí dự phòng thấp hơn, ANPR thường là một lựa chọn trong các doanh nghiệp thu phí và bãi đậu xe. Lý do chính là hệ thống ANPR nhận dạng biển số đã đăng ký mà không có thêm yêu cầu nào. Đô thị hóa nhanh chóng của các quốc gia là một bước tiến vượt bậc trong thế giới hiện đại. Người dân di cư khỏi các vùng nông thôn và chủ yếu chọn sống ở các thành phố. ANPR ngày càng được sử dụng nhiều hơn để kiểm tra mức độ tự do của giao thông, tạo điều kiện thuận lợi cho giao thông thông minh [1]. Máy ảnh ANPR hiện đại không chỉ có thể đọc biển số, mà chúng có thể cung cấp thông tin hữu ích như đếm, phương hướng, loại phương tiện và tốc độ của chúng. Với khả năng phát hiện và đọc khối lượng lớn các phương tiện di chuyển nhanh, nó được sử dụng trong các ứng dụng rất đa dạng như kiểm soát ra vào, quản lý bãi đậu xe, tính tiền, thanh toán người dùng, theo dõi giao hàng, quản lý đường đi, dịch vụ chính sách và an ninh, dịch vụ khách hàng và chỉ đường, thực thi đèn đỏ và làn đường, ước tính độ dài hàng đợi, và nhiều dịch vụ khác [2–8].



II. CÔNG TRÌNH LIÊN QUAN

Trong [1-2], một mô hình học sâu được tạo ra để nhận dạng biển số bằng cách sử dụng tập dữ liệu Thổ Nhĩ Kỳ do họ tạo ra. Họ đã sử dụng khung Tensorflow với thư viện học sâu Keras. Họ đã thu thập 34, 58 hình ảnh trong đó thuật toán smearing được áp dụng bằng chương trình MATLAB. 75% hình ảnh được sử dụng để đào tạo, 25% để thử nghiệm và 5% để xác nhận. Vì hình ảnh được chụp từ nền thời gian thực, họ đã thực hiện một số kỹ thuật xử lý hình ảnh như làm mịn độ mờ trung bình, ngưỡng Gauss thích ứng và phép biến đổi hình thái. Sau những bước chuẩn bị này, mẫu CNN được đào tạo bằng cách sử dụng các hình ảnh. Các đặc trưng ảnh trích xuất từ CNN được áp dụng cho mạng LSTM theo sau là thuật toán decryption.

[3], một hệ thống nhận dạng biển số tự động sử dụng phương pháp học máy. Họ nhận input từ camera hồng ngoại, sau đó là tăng độ tương phản và giảm nhiễu như các bước tiền xử lý. Sau đó, họ khoanh vùng biển số trong ảnh bằng cách tìm ra RoI. Sau đó, theo dõi đường viền được áp dụng để có được các đặc trưng nổi bật của hình ảnh. Sau đó, tính năng dò tìm cạnh của Canny được thực hiện để tìm ra các cạnh của các ký tự trong biển số. Cuối cùng, phân đoạn được áp dụng để tách các ký tự. Các ký tự riêng lẻ được nhận dạng bằng cách sử dụng đối sánh mẫu bởi Artificial Neural Networks (ANN). Toàn bộ hệ thống được phát triển bằng phần mềm MATLAB.

Một cuộc khảo sát đã được thực hiện [4] về các phương pháp khác nhau được sử dụng để triển khai Nhận dạng biển số tự động (ANPR). Các bước cơ bản gồm Chụp ảnh phương tiện, phát hiện biển số, phân đoạn ký tự và nhận dạng ký tự. Để phát hiện biển số, các yếu tố như kích thước tâm, vị trí tâm, nền tâm và vết phải được xem xét. Độ chính xác tối đa để phát hiện tâm đạt được bằng Phát hiện cạnh của Canny. Phân

đoạn ký tự có thể được thực hiện bằng cách sử dụng binarization hình ảnh, CCA (Connected Component Analysis), phép chiếu dọc và ngang tạo ra kết quả tốt hơn. Tiếp theo là nhận dạng ký tự thường được thực hiện bằng Mạng thần kinh nhân tạo, đối sánh mẫu hoặc kỹ thuật Nhận dạng ký tự quang học (OCR).

Trong [5], đánh giá dựa trên Nhận dạng biến số tự động (ANPR) được thực hiện. Hệ thống được đề xuất bao gồm một mô-đun camera, cảm biến, bộ điều khiển, GSM và một máy chủ nhúng. Nó cố gắng chặn các phương tiện trái phép bằng cách so sánh cơ sở dữ liệu phương tiện đã được lưu. Hình ảnh được chụp từ máy ảnh được chuyển sang thang xám và được nâng cao bằng cách điều chỉnh biểu đồ. Các cạnh được phát hiện bằng cách sử dụng phương pháp phát hiện cạnh của Sobel. Sau đó, xử lý hình thái học(morphological) được thực hiện. Sau đó, phân đoạn được thực hiện trên hình ảnh được phát hiện cạnh. Cuối cùng, các ký tự được nhận dạng bằng cách sử dụng phương pháp học máy.

Trong [6], một hệ thống nhận dạng tấm sử dụng phương pháp học sâu được phát triển. Họ đã phát triển một hệ thống OCR với một bộ dữ liệu tùy chỉnh. Tập dữ liệu được tạo ra một cách nhân tạo bằng cách lấy một số hình ảnh từ internet và thêm tiếng ồn và hình nền cho những hình ảnh đó. Đối với nền, cơ sở dữ liệu SUN và cơ sở dữ liệu Standford được sử dụng. Để phát hiện biến số YOLO, một khung phát hiện đối tượng được sử dụng. Để nhận dạng ký tự, Mạng nơ-ron tích chập (CNN) được sử dụng. Lớp đầu ra của CNN bao gồm 7,62 tỷ bào thần kinh cho 7 ký tự. Xác nhận chéo 10 lần được áp dụng ở đầu ra để tìm độ chính xác.

Trong [7], một hệ thống nhận dạng biến số được tạo ra bằng cách sử dụng Mạng nơ-ron hợp pháp. Hình ảnh chụp từ máy ảnh được xử lý trước bằng cách chuyển đổi hình ảnh RGB sang ảnh xám, loại bỏ nhiễu và mã hóa nhị phân. Sau đó, biến số xe được trích xuất bằng cách sử dụng phương pháp Connected Component tùy thuộc vào các thuộc tính như độ dài trục chính, độ dài trục phụ, diện tích và bounding box,... Các ký tự trong biến số xe được trích xuất được phân đoạn bằng cách quét ngang và quét dọc. Cuối cùng, các ký tự được nhận dạng bằng cách sử dụng mạng nơ-ron tích chập (CNN). Tập dữ liệu được sử dụng để đào tạo CNN bao gồm 1000 hình ảnh cho mỗi 36 ký tự. Trong số 36.000 hình ảnh, 30.000 mẫu được sử dụng làm dữ liệu đào tạo và 6000 mẫu để thử nghiệm. Họ đã sử dụng thuật toán giảm thiểu để giảm thiểu entropy chéo với tỷ lệ học tập là 0,5.

[9], một hệ thống tự động nhận dạng biến số xe được phát triển. Hệ thống được chia thành ba loại: phát hiện biến số xe, phân đoạn ký tự và nhận dạng ký tự. Trong phát hiện biến số xe, ảnh input được chuyển thành ảnh HSV và được áp dụng với một số bộ lọc. Sau đó, CNN được áp dụng trên các hình ảnh để phát hiện các tấm biển. Trong phân đoạn ký tự, các hình ảnh thông qua bước tiền xử lý. Mô hình CNN thứ hai được áp dụng để phân đoạn các ký tự trong ảnh. Cuối cùng, lớp softmax được áp dụng để xác định xác suất lớp. Tập dữ liệu được sử dụng có hai lớp: lớp tích cực và lớp tiêu cực. Tập dữ liệu đầu tiên là tập dữ liệu LP và tập dữ liệu không phải LP. Tập dữ liệu thứ hai dành cho ký tự và không phải ký tự.

Thuật toán phát hiện đối tượng YOLO được phát triển bằng cách sử dụng mạng nơ-ron tích chập. CNN tăng hiệu quả và bằng cách tăng các lớp ẩn, hệ thống sẽ học tốt về mối quan hệ giữa các lớp đầu vào và đầu ra. Hạn chế chính trong YOLO là, nó hoạt động tốt để phát hiện video trực tiếp. Nhưng để xử lý mọi khung hình, quá trình này rất dài. Trong kiến trúc mạng nơ-ron, độ phức tạp của code rất cao và cần có GPU để xử lý mạng nơ-ron, nhưng nó mang lại hiệu quả cao khi hệ thống được đào tạo với nhiều lớp hơn. Để giảm độ phức tạp của code và cũng để giảm tốc độ xử lý, khung ImageAI được sử dụng và nó tạo ra hiệu quả cao hơn so với các phương pháp khác đã thảo luận trước đó.

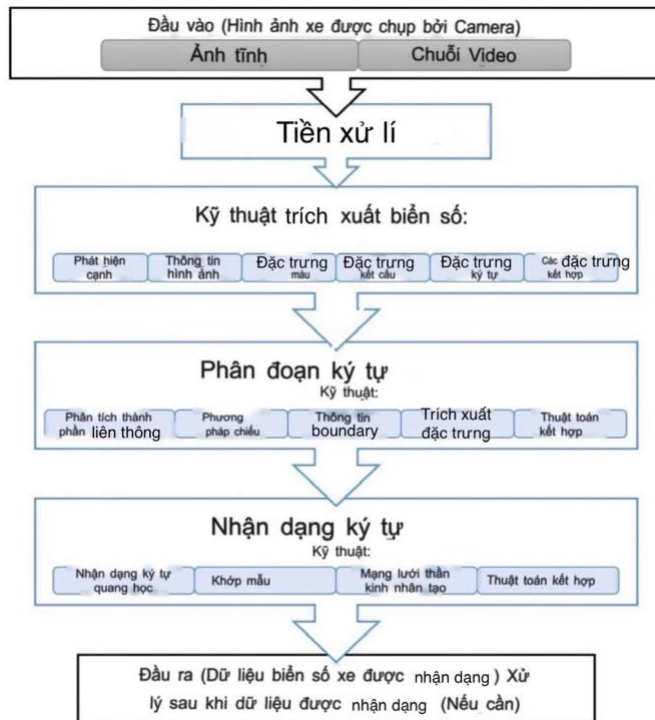
Phương pháp	Trích xuất biến số	Phân đoạn ký tự	Nhận dạng ký tự	Hiệu năng	Datas
[1][2]	Thuật toán smearing	Ngưỡng Gaussian, Phép biến đổi hình thái	CNN, LSTM dùng thuật toán decryption	Khung 96.36% Số 99.43% Chữ 99.05% Kí tự 99.31%	Turki
[3]	Tìm vùng có đặc trưng RoI, Phát hiện biên cạnh Canny	-	Đối sánh pattern bằng ANN	-	-
[4]	Canny	Binarization, phân tích thành phần liên thông CCA, chiếu dọc chiếu ngang	ANN, đối sánh template, kĩ thuật OCR	98.7%	-
[5]	Sobel	Xử lí ảnh hình thái học morphology	máy học	-	-
[6]	Yolo	-	CNN	Khung 98.5% Kí tự 96.8% 94%	Sun, Stand
[7]	Thuật toán thành phần liên thông Connected Component	Scan dọc scan ngang	CNN	97%	-
[9]	CNN	-	CNN	94.8%	LP, no LP.
Đề xuất	Yolo	phân tích thành phần liên thông CCA	CNN	-	Ảnh tự lấy.

III. PHÁT BIỂU BÀI TOÁN

Input: Hình ảnh/video có chứa biển số.

Output: Vị trí bounding box của khung biển số và class id là thông tin chữ và số có trong khung biển số ở input.

Cơ chế hoạt động (framework):



Thách thức

Quá trình training trải qua nhiều công đoạn.

Biển số 1 dòng và 2 dòng.

IV. CÁC TẦNG CƠ BẢN CỦA HỆ THỐNG THỊ GIÁC

1. Tầng 1 – Xử lý dữ liệu cơ bản

a. Tiền xử lý

Phát hiện biên cạnh.

b. Rút trích đặc trưng

Bước 1: Đầu tiên, khởi tạo ngẫu nhiên tham số cho những bộ lọc.

Bước 2: Khi cho ảnh qua bộ lọc, ta được một số giá trị đầu ra output.

Bước 3: Sau đó tiếp tục cho output qua các lớp tiếp theo trong mạng nơ-ron và mạng nơ-ron tiên đoán nhãn cho tấm này.

Bước 4: Máy tính sẽ so sánh nhãn này với nhãn đúng của tập ảnh trong tập huấn luyện và điều chỉnh tham số của bộ lọc để kết quả gán nhãn giống hơn với nhãn đúng.

Bước 5: Quá trình lặp đi lặp lại và mạng nơ-ron sẽ học được cách gán nhãn tốt nhất có thể.

c. Kiến trúc mạng

Mạng học sâu tích chập (Deep Convolutional Neural Networks). Ý tưởng là áp dụng các bộ lọc lên ảnh trước khi huấn luyện mạng nơ-ron. Xong, các đặc trưng của tấm ảnh sẽ trở nên nổi bật và ta có thể dùng chúng để nhận dạng hình ảnh. Bộ lọc thực chất chỉ là tổ hợp của những phép nhân và việc chọn giá trị cho bộ lọc chính là chìa khóa của kiến trúc mạng này.

d. Mô hình học

Đối với chủ đề này, nhóm dùng mô hình học có giám sát - là thuật toán tiên đoán nhãn cho dữ liệu mới dựa trên tập huấn luyện (các mẫu trong tập này đều đã được gán nhãn). Thông qua quá trình huấn luyện, một mô hình sẽ được xây dựng cho các tiên đoán và khi các tiên đoán bị sai thì mô hình này sẽ được tinh chỉnh lại. Việc huấn luyện sẽ dừng lại khi mô hình đạt đến độ chính xác mong muốn dựa trên dữ liệu huấn luyện.

2. Tầng 2 – Xử lý các tác vụ đơn

a. Phát hiện đối tượng

Ở chủ đề này, nhóm sẽ sử dụng Yolov3, thích hợp cho bài toán phân lớp.

b. Phân lớp đối tượng

Phân lớp (classification) là một kỹ thuật nhằm phân tích các dữ liệu (observation), sau đó phân bố các đối tượng mới (observation) cho các class khác nhau đã được xác định trước đó. Bài toán phân lớp gồm hai nhiệm vụ chính: Thứ nhất, sắp xếp các đối tượng (observation) vào 2 hay nhiều lớp đã được dán nhãn. Thứ hai, tìm ra một phương pháp tối ưu để gán nhãn đối tượng cho các lớp đã được gán nhãn trước.

c. Nhận dạng đối tượng

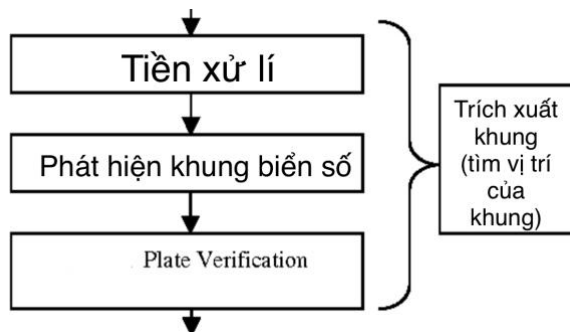
Khái niệm nhận dạng thực chất là quá trình phát hiện và phân lớp đối tượng.

3. Tầng 3 – Xử lý các ứng dụng phức hợp

Nhận dạng biển số để làm gì? Nhu cầu để giám sát an ninh, bãi đỗ xe thông minh, hệ thống thu tiền không dừng tại các tuyến đường cao tốc...

V. FLOW CHART CÁC KỸ THUẬT

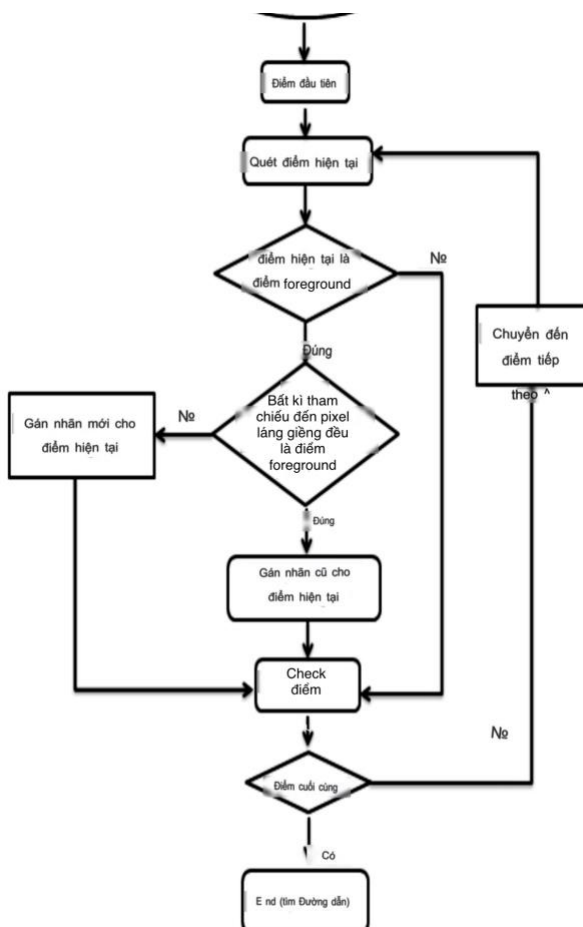
1. Trích xuất biên số



2. Phân đoạn kí tự

Phân đoạn là quá trình vẽ boundary bao trọn đối tượng, hơn là chỉ qua quét đóng vị trí bounding box của đối tượng.

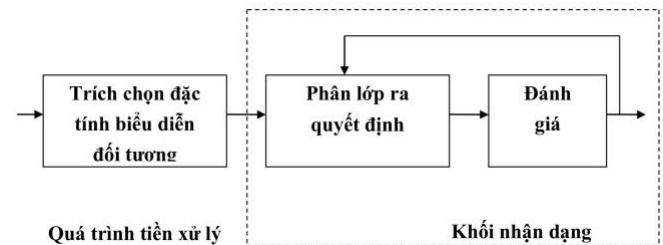
Connected components analysis



Thuật toán này có ý tưởng đơn giản là nó sẽ kết nối tất cả pixel nào có cùng giá trị thành một khối và gán cho nó một cái nhãn(label). Nhờ đó tất cả các pixel của cùng một kí tự

và có cùng giá trị sẽ được kết nối và được tách ra khỏi biên số xe.

3. Nhận dạng kí tự



VI. DATASET

Yolov3

<https://drive.google.com/drive/folders/1em3nTEpu3akZDcTtNefFGQJzhvMjU3JE?usp=sharing>

Yolov4

https://drive.google.com/drive/folders/1Z5NNnJttKwgPCGSDyOtQ1QIY-h_LGHf3?usp=sharing

VII. MÔI TRƯỜNG

Công cụ: Google Colab, Visual Studio Code

Xem cách cài venv cho Visual Studio Code tại [11]

Ngôn ngữ: Python

VIII. THỰC NGHIỆM

Thực nghiệm với Yolov3.

Một số lệnh cần chạy trước ở Terminal:

```

pip install tensorflow
pip install imutils
pip install keras
pip install --upgrade setuptools pip wheel
pip install -r /Users/jason/Downloads/License-Plate-Recognition-master/requirements.txt

pip install PyQtWebEngine==5.12.1

pip install PyQt5==5.12.3
  
```

File .weights (yolov3) trong /src/weights:

<https://drive.google.com/file/d/1IUvj-6qPE25Tn-cSnEL6zq8zpWgidQo/view?usp=sharing>

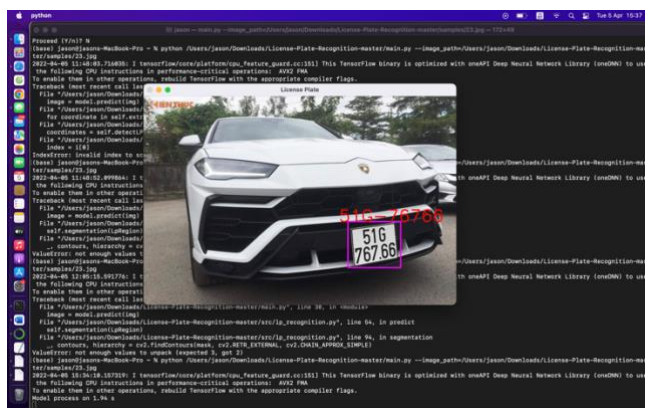
File weight.h5 trong /src/weights:

https://drive.google.com/file/d/1Q_iScnPA7Zzl1KBDus499Mvvr9fk3w82/view?usp=sharing

Ảnh kết quả theo câu lệnh sau:

```
python '/Users/jason/Downloads/License-Plate-Recognition-master-3/main.py' --image_path='/Users/jason/Downloads/License-Plate-Recognition-master-3/samples/23.jpg'
```

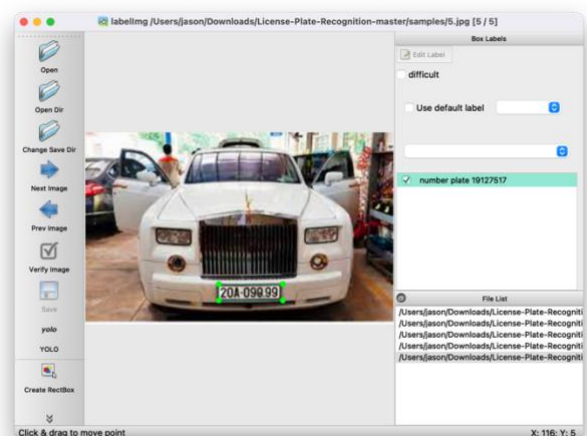
(--image_path= ảnh trong dataset yolov3)



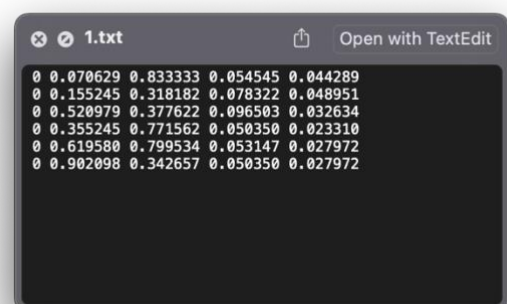
```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.all.sh)"
brew install python3
pip3 install pipenv
pipenv run pip install pyqt5==5.12.1 lxml
pipenv run make qt5py3
cd <thư mục labellmg>
pyrcc5 -o libs/resources.py resources.qrc
python3 labellmg.py
```

Giai đoạn 1: Training

Hướng dẫn training xem tại [9] và [10].



Gán nhãn number plate 19127517



.txt ứng với mỗi ảnh thu được sau khi dùng labellmg

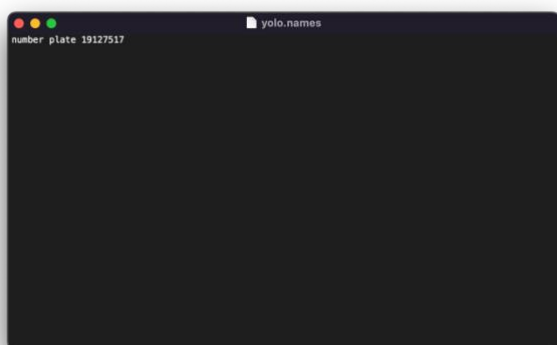
IX. CẢI TIẾN

Thực nghiệm với Yolov4.

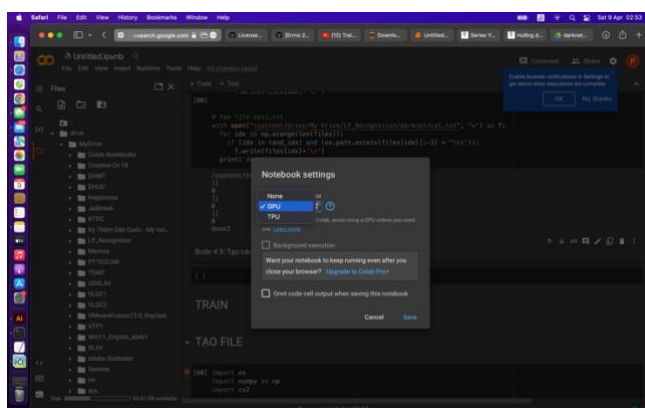
Một số lệnh cần chạy trước khi dùng tool labellmg ở Terminal:



Tạo yolo.data trong /content/darknet

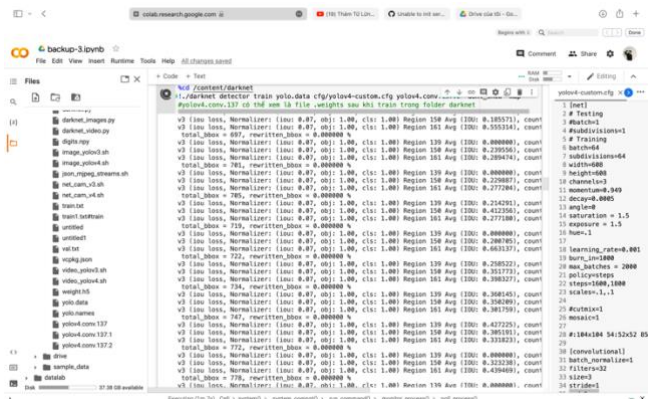


Tạo yolo.names trong /content/darknet



Runtime GPU trong Google Colab

Chi tiết quá trình training của chúng em ghi rất cụ thể ở file .ipynb đính kèm. Quá trình training diễn ra tầm 4-5 tiếng. Điều kiện dừng khi Average Loss thuộc khoảng [0;1]. Sau khi train ta được yolov4-custom_last-3.weights



Output khi train thành công

File /content/darknet của nhóm

https://drive.google.com/drive/folders/12kIdJ0bfqIMHbpTVqjCspN8bPrbI2_WE?usp=sharing

File .weights của nhóm (yolov4) trong thư mục /src/weights:

https://drive.google.com/file/d/1bWtLP0-61C5_qa2sdSdrFWee18tvtY/view?usp=sharing

File yolov4.conv.137.2 để ở /content/darknet

<https://drive.google.com/file/d/1wXw12v2Ev4oBbt1rpBVg3Ot1vq4keSV2/view?usp=sharing>

File weight.h5 trong /src/weights:

https://drive.google.com/file/d/1Q_iScnPA7Zz11KBDus499Mvv9f3k3w82/view?usp=sharing

Giai đoạn 2: Lấy weights sau training đem đi nhận dạng

Mẹo để đỡ phải chỉnh sửa lại path trong source code:

1. Ghi đè yolo.names của yolov4 lên yolo.names của yolov3.

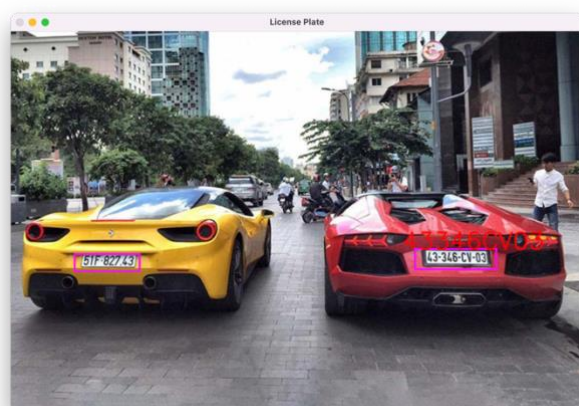
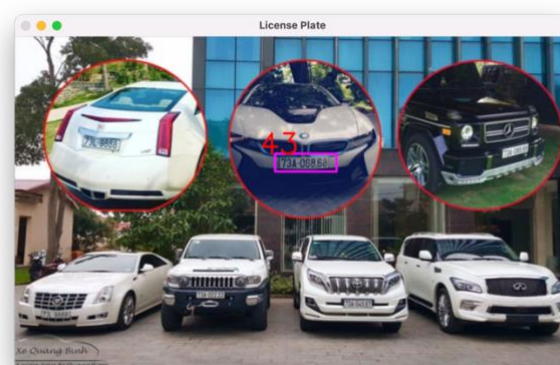
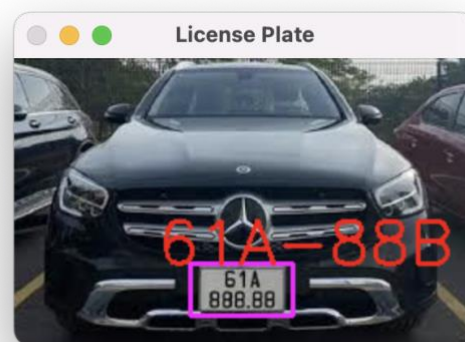
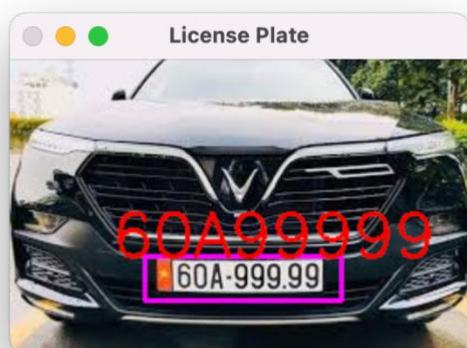
2. Ghi đè yolov4-custom_last-3.weights của yolov4 lên yolov3-tiny_15000.weights của yolov3 và rename tên giống với của yolov3.

3. Ghi đè yolov4-tiny-custom.cfg của yolov4 lên yolov3-tiny.cfg của yolov3 và rename tên giống với của yolov3.

Theo câu lệnh sau, ta được ảnh kết quả (--image_path= là đường dẫn ảnh của dataset yolov4):

```

python /Users/jason/Downloads/License-Plate-Recognition-master-3/main.py
image_path='/Users/jason/Downloads/darknet' --content-darknet/data/12.jpeg'
  
```



X. NHẬN XÉT

Yolov3

Hệ thống có thể hoạt động trên cả biển một dòng hoặc hai dòng. Thậm chí đôi khi biển số xe bị che khuất một chút vẫn đọc được. Tuy nhiên, nó vẫn có một số nhược điểm: Một, Khi ảnh đầu vào bị đặt một góc quá nghiêng thì một vài kí tự sẽ bị nhầm dòng. Có một cách giải quyết là dùng một mạng transformer xoay ảnh nghiêng về ảnh thẳng. Hai, Đôi khi bị nhận dạng nhầm giữa 8 và B, 0 và D. Ba, Hoạt động kém khi bức ảnh quá mờ.

Yolov4

Hệ thống còn lỗi nhận dạng sai vì dataset ít và vòng lặp chưa đủ lớn. Ngoài ra, chúng ta có thể tìm google và tải các dataset lớn hơn đã gán nhãn sẵn. Ở đây quá trình training chỉ diễn ra 4-5 tiếng, chúng ta có thể train lâu hơn với 1000 vòng lặp để tăng độ chính xác.

XI. TỔNG KẾT

Qua bài thực hành này, chúng em có thêm kiến thức về việc áp dụng các bài học lý thuyết trên lớp vào những vấn đề thực tế. Chúng em biết thêm cách train dữ liệu trên Google Colab với runtime GPU. Ngoài ra còn biết thêm cách sử dụng Visual Studio Code để code Python, cách dùng tool labellmg để gán nhãn đối tượng. Tuy quá trình training dài nhưng đây đều là những công đoạn cơ bản không quá phức tạp.

Xung quanh còn rất nhiều đề án khác (cũng có bài thi cuối kỳ) nên thời gian dành cho đề án này không đủ để phát triển thêm. Trong phạm vi đề án, chúng em cũng đã thử các tổng hợp nhiều kiến thức nhất có thể từ bài giảng và tham khảo thêm nhiều tư liệu khác. Để hoàn thành được bài thực hành này, chúng em xin cảm ơn các giảng viên đã hỗ trợ nhiệt tình, tận tâm hết lòng vì chúng em trong môn học này, kính chúc thầy cô sức khỏe và niềm vui trong công việc giảng dạy tại HCMUS.

Lời nói cuối cùng, mong sao ta sẽ được gặp lại.

XII. THAM KHẢO

[1] İrfan Kılıç, Galip Aydın 2018 Turkish Vehicle License Plate Recognition Using Deep Learning International Conference on Artificial Intelligence and Data Processing pp.1-5.

[2] J. V. Bagade Sukanya Kamble Kushal Pardeshi Bhushan Punjabi Rajpratap Singh Automatic Number Plate Recognition System: Machine Learning Approach IOSR Journal of Computer Engineering pp. 34-39.

[3] Atul Patel Chirag Patel Dipti Shah 2013 Automatic Number Plate Recognition System (ANPR): A Survey International Journal of Computer Applications Volume 69–No.9 pp. (0975 – 8887).

[4] Shraddha S Ghadage Sagar R Khedkar 2019 A Review Paper on Automatic Number Plate Recognition System using Machine Learning Algorithms International Journal of Engineering Research & Technology (IJERT) Vol. 8 Issue 12.

[5] Marko Arsenovic Srđan Sladojevic Andras Anderla Darko Stefanovic 2017 Deep Learning Driven Plates Recognition System XVII International Scientific Conference on Industrial Systems Novi Sad Serbia.

[6] Shrutika Saunshi Vishal Sahani Juhi Patil Abhishek Yadav Dr. Sheetal Rathi License Plate Recognition Using Convolutional Neural Network IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727.

[7] Selmi Zied & Ben Halima Mohamed & Alimi Adel 2017 Deep Learning System for Automatic License Plate Detection and Recognition 14th IAPR International

Conference on Document Analysis and Recognition (ICDAR) pp. 1132-1138.

[8] Lubna, Naveed Mufti and Syed Afaq Ali Shah Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms. Sensors 2021, 21, 3028.

[9] training <https://devai.info/2020/12/15/huong-dan-training-object-detection-voi-yolov4-su-dung-google-colab/>

[10] training <https://www.youtube.com/watch?v=kIv8dwDGKkw>

[11] VSCode và venv <https://courses.ctda.hcmus.edu.vn/mod/resource/view.php?id=51436>