ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

ĐẶNG THỊ MỸ NHÀN

ỨNG DỤNG DEEP LEARNING ĐỂ ĐẾM SỐ LƯỢNG XE ÔTÔ TRONG NỘI THÀNH ĐÀ NẴNG

Chuyên ngành: Khoa Học Máy Tính

Mã số: 8480101

LUẬN VĂN THẠC SĨ

NGƯỜI HƯỚNG DẪN KHOA HỌC: **TS. TRẦN THẾ VŨ**

Đà Nẵng - Năm 2019

Công trình được hoàn thành tại

TRƯỜNG ĐẠI HỌC BÁCH KHOA

Người hướng dẫn khoa học: TS. TRẦN THẾ VŨ

Phản biện 1: TS. Nguyễn Văn Hiệu

Phản biện 2: TS. Phạm Xuân Hậu

Luận văn đã được bảo vệ trước Hội đồng chấm luận văn tốt nghiệp thạc sĩ Khoa học máy tính tại Trường Đại học Bách khoa vào ngày 25 tháng 8 năm 2019.

Có thể tìm hiểu luận văn tại:

- Trung tâm Học liệu, Đại học Đà Nẵng tại Trường Đại học Bách khoa.

- Thư viện Khoa Công nghệ thong tin, Trường Đại học Bách khoa - Đại học Đà Nẵng.

PHẦN I: MỞ ĐẦU

1. Lý do chọn đề tài

Giao thông luôn là vấn đề được quan tâm nhiều nhất đối với các đô thị lớn ở các nước nói chung và ở Việt Nam nói riêng. Trong những năm gần đây tình trạng kẹt xe vẫn luôn là vấn đề nghiêm trọng và bức thiết nhất. Đây là vấn đề bức xúc của toàn xã hội, ảnh hưởng đến sự phát triển bền vững về kinh tế, văn hóa, xã hội và hình ảnh của đất nước Việt Nam với bạn bè Quốc tế.

Hiện nay, mặc dù đã có những nghiên cứu, nhiều giải pháp tháo gỡ tình trạng tắc nghẽn giao thông nhưng hiệu quả chưa cao và tình trạng kẹt xe vẫn xảy ra thường xuyên trên các trục đường giao thông, đặc biệt là ở các khu đô thị.

Xuất phát từ yêu cầu thực tiễn tôi đã chọn đề tài "Ứng dụng Deep Learning để đếm số lượng xe ô tô trong nội thành Đà Nẵng".

2. Mục tiêu và nhiệm vụ

- *Mục tiêu*

Từ thực trạng trên đòi hỏi mục tiêu đặt ra trước mắt là:

- Giảm thiểu ùn tắc, tai nạn giao thông.
- Nâng cao nhận thức và ý thức tự giác chấp hành giao thông .
- Nghiên cứu và nhân rộng các giải pháp đột phá về khắc phục tình trạng ùn tắc giao thông, đồng thời và nâng cao hiệu quả quản lý trật tự đô thị.

- *Nhiệm vụ*

Từ mục tiêu của đề tài, luận văn tập trung nghiên cứu các nhiệm vụ sau:

- Tìm hiểu kỹ thuật Deep Learning.
- Tìm hiểu các bộ thư viện cài đặt cho mô hình Deep Learning.

- Sử dụng công cụ nhận diện để đánh nhãn ôtô trong video giao thông.
- Huấn luyện và xây dựng mô hình nhận dạng, đếm xe ôtô.

3. Ý nghĩa khoa học và thực tiễn

- Về khoa học: Áp dụng phương pháp Deep Learning để phát hiện, tính được tốc độ và đếm xe ô tô.
- Về thực tiễn: Hỗ trợ xác định lưu lượng ô tô đang tham gia giao thông trên các trục đường giao thông, giúp giải quyết các vấn đề về giao thông.

4. Bố cục luận văn

- Chương 1: Giới thiệu khái niệm về giao thông và thực trạng giao thông Việt Nam; giới thiệu khái niệm về xử lý ảnh, học máy, bài toán nhận dạng ô tô qua hình ảnh và tổng quan các kết quả nghiên cứu liên quan ở trong nước và ngoài nước.
- Chương 2: Giới thiệu về môi trường và công cụ sử dụng bao gồm thư viện mã nguồn Tensorflow Object Detection, thuật toán SSD (Single Shot Object Detectors), thuật toán Faster R-CNN, mô hình Object Detectors; Các bước phát triển chương trình bao gồm 5 bước; Thu thập dữ liệu, đánh nhãn dữ liệu, đào tạo mô hình, nhận dạng và đếm phương tiện.
- Chương 3: Giới thiệu kết quả thực nghiệm và đánh giá kết quả nhận dạng và đếm số lượng phương tiện trên một video mẫu được trích xuất từ camera giao thông tại một đoạn đường cụ thể trong nội thành Đà Nẵng.
- Kết luận và hướng phát triển.

PHẦN II: NỘI DUNG

CHƯƠNG 1. NGHIÊN CỨU TỔNG QUAN

1.1. KHÁI NIỆM VÀ THỰC TRẠNG GIAO THÔNG VIỆT NAM

1.1.1. Khái niệm về giao thông

1.1.2. Thực trạng giao thông Việt Nam

1.2. KHÁI NIỆM VỀ XỬ LÝ ẢNH, HỌC MÁY VÀ BÀI TOÁN NHẬN DẠNG Ô TÔ QUA HÌNH ẢNH

1.2.1. Khái niệm về xử lý ảnh

Xử lý ảnh là phương pháp chuyển đổi hình ảnh sang dạng số và thực hiện một số hoạt động trên đó để nâng cao chất lượng hình ảnh hoặc để trích xuất một số thông tin hữu ích từ nó.

1.2.2. Tổng quan về Học máy

Học máy - Machine Learning là một bộ phận của của trí tuệ nhân tạo (AI), các hệ thống mà sau khi được cung cấp một lượng dữ liệu và thực hiện một hoặc một số thao tác dựa trên những dữ liệu đã được cung cấp để máy có thể học. Machine Learning thường được dùng trong việc xử lý các tác vụ tương tự con người mà khó có thể mô phỏng thành công thức cụ thể, ví dụ như nhận diện, đánh giá lựa chọn, ...

a, Các phương thức Machine Learning

Các thuật toán Machine Learning thường được chia làm 4 nhóm:

- Học có giám sát (dạy học)
- Học không giám sát (tự học)
- Học bán giám sát (vừa học vừa tự học)
- Reinforcement Learning (học tăng cường)

b, Các nhóm Machine Learning dựa trên chức năng

- Thuật toán phân loại (Classification Algorithms) bao gồm: Phân loại tuyến tính (Linear Classifier), Máy hỗ trợ vector (Support Vector Machine – SVM).
- Giải thuật dựa vào thể thức (Instance-based Algorithms): K-Nearest Neighbor (KNN), Learning Vector Quantization (LVQ).
- Giải thuật Bayesian: Naive Bayes, Gaussian Naive Bayes
- Giải thuật phân cụm: K-Means clustering, K-Medians.
- Dimensionality Reduction Algorithms: Phân tích các thành phần độc lập (Principal Component Analysis –PCA), Phân loại tuyến tính (Linear Discriminant Analysis – LDA).

1.2.3. Bài toán nhận dạng ô tô qua hình ảnh

a. Giới thiệu về bài toán nhận dạng ô tô qua hình ảnh

Những năm gần đây, có rất nhiều công trình nghiên cứu về bài toán nhận dạng ô tô. Ban đầu chỉ là những bài toán đơn giản, mỗi ảnh chỉ có một hình ảnh xe ô tô với góc chụp chính diện, không đáp ứng được nhu cầu ngày càng cao trong cuộc sống, khoa học ngày nay. Vì thế đã có những cải tiến nghiên cứu về bài toán phát hiện xe ô tô trong những môi trường phức tạp hơn, có nhiều xe ô tô xuất hiện đồng thờivới nhiều góc chụp khác nhau cùng với các phương tiện khác trong ảnh.

Phát hiện xe ô tô là một kỹ thuật để phát hiện vị trí, màu sắc và kích thước khác nhau trong các ảnh bất kỳ. Kỹ thuật này chỉ nhận biết về các đặc trưng của xe ô tô và bỏ qua những đặc trưng khác.

b. Các phương pháp chính nhận dạng xe ô tô

Dựa vào tính chất của các phương pháp xác định xe ô tô trên ảnh, các phương pháp này được chia thành bốn loại chính, tương ứng với bốn hướng tiếp cận khác nhau.

- Hướng tiếp cận dựa trên tri thức:

- Hướng tiếp cận dựa trên đặc trưng không thay đổi.
- Hướng tiếp cận dựa trên so sánh khớp mẫu.
- Hướng tiếp cận dựa trên ảnh mẫu có sẵn.

c. Những khó khăn khi nhận dạng xe ô tô trong ảnh

- Xe ô tô trong ảnh có thể có những góc chụp khác nhau.
- Mỗi loại xe có những đặc trưng khác nhau.
- Một số xe bị che khuất bởi các đối tượng.
- Giới hạn về số hình ảnh cần thiết cho việc huấn luyện.

1.3. TỔNG QUAN CÁC KẾT QUẢ NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC

1.3.1. Một số các nghiên cứu đã và đang được thực hiện tại nước ta

- Nghiên cứu phát hiện làn đường, ôtô và người đi bộ bằng công nghệ ảnh hỗ trợ cho ôtô tự hành của hai tác giả Trương Quốc Bảo, Trương Quốc Định - Trường ĐH Cần Thơ [1].
- Các nhà khoa học thuộc Khoa Điện - Điện tử, Trường Đại học Giao thông Vận tải đã nghiên cứu thiết kế và chế tạo thành công hệ thống giám sát giao thông ứng dụng công nghệ xử lý ảnh. Hệ thống bao gồm camera giám sát, camera chụp hình, mạng truyền thông, video server, phần mềm xử lý ảnh và cơ sở dữ liệu có thể tự động phát hiện và ghi nhận các tình huống vi phạm Luật Giao thông nhằm tăng cường giám sát, phát hiện và xử lý kịp thời các vi phạm, hạn chế tai nạn và nâng cao ý thức chấp hành của người tham gia giao thông.

1.3.2. Các kết quả nghiên cứu ngoài nước

➢ *Nghiên cứu của ông Andrews Sobral:*

Ông Andrews Sobral và cộng sự đã nghiên cứu nhận diện phương tiện qua phương pháp Haar Cascades using OpenCV [3]. Nghiên cứu sử

dụng thư viện thị giác máy tính mã nguồn mở JavaScript và OpenCV để nhận diện hình ảnh được truy cập vào webcam.

- *Nghiên cứu nhận diện và theo dõi xe sử dụng OpenCV và Kalman Filter*

Ông Ronit Sinha và cộng sự sử dụng OpenCV và Kalman Filter để nhận diện và theo dõi xe từ luồng trực tuyến của Camera giao thông [4]. Bộ lọc OpenCV và Kalman sẽ phát hiện và theo dõi xe ô tô từ video được phát trực tiếp từ camera.

- *Nghiên cứu về nhận diện vật thể bằng thuật toán YOLO:*

Yolo được Redmon và Farhadi phát triển vào năm 2015, trong thời gian học tiến sĩ [5]. YOLO ('You only look once") là một thuật toán nhận diện phổ biến nhờ độ chính xác cao trong thời gian thực, đạt đến 45 khung hình trên giây.

- *Nghiên cứu nhận diện phương tiện và hướng đi (Vehicle Detection for Autonomous Driving)*

Ông Junsheng Fu và cộng sự đã sử dụng các công cụ, bao gồm OpenCV3, Python3.5, tensorflow, CUDA8 OS: Ubuntu 16.04 [6].Có 2 hướng đi: SVM tuyến tính và mạng Neural.

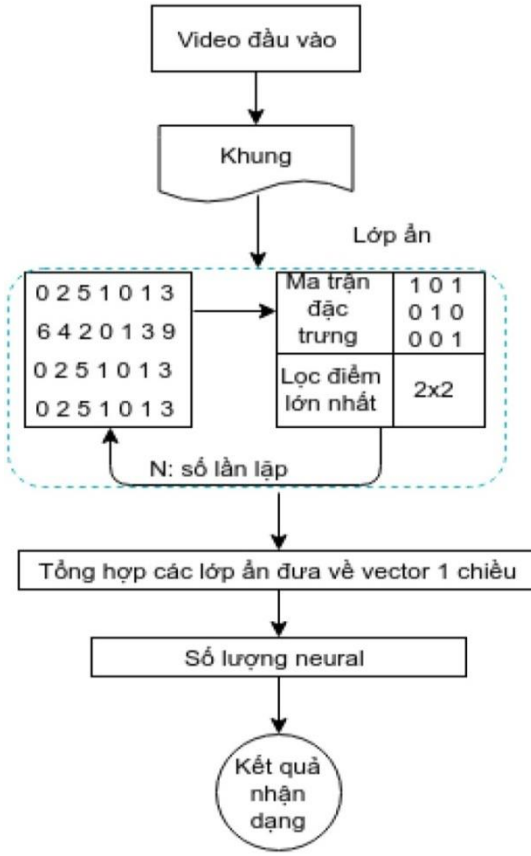
CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

2.1. NHẬN DẠNG ĐỐI TƯỢNG XE Ô TÔ DỰA VÀO DEEP LEARNING

Mục tiêu của bài toán là từ một video cụ thể thì phải nhận diện ra được các phương tiện xuất hiện trong video đó.

Sau khi đã nhận diện được các đối tượng xe, chúng ta tiến hành đếm xe. Yêu cầu đặt ra cho bài toán đó là tính chính xác, sai số không đáng kể.

Một trong những giải pháp đó là ứng dụng Deep Learning để nhận diện xe.



[Hình 2.1] Sử dụng Deep Learning nhận dạng xe ô tô

Deep Learning được huấn luyện bằng cách sử dụng một tập hợp dữ liệu lớn có gắn nhãn và kiến trúc mạng (thần kinh) chứa nhiều lớp, điều này đã giúp máy tính thực thi những việc như: Phân loại cả ngàn vật thể khác nhau trong các bức ảnh; Tự tạo chú thích cho ảnh; Bắt chước giọng nói và chữ viết của con người; Giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc.

Deep Learning sử dụng phương pháp Neural Network với nhiều tầng (layer) khác nhau mà các tầng đều là các hàm phi tuyến (hàm đa thức).

Deep Learningcó nhiều điểm nổi bật hơn so với phương pháp Machine Learning:sử dụng nhiều lớp nên có khả năng "mô hình hóa" những bài toán có độ phức tạp cao, cần một lượng lớn dữ liệu để huấn luyện.Thông thường đầu vào của một hệ thống Deep Learning là dữ liệu thô, còn đầu vào của Machine Learning là dữ liệu đã được trích xuất.Deep Learning có phương pháp huấn luyện và đào tạo để thực hiện phân loại đối tượng tốt hơn Machine Learning với 3 phương pháp phổ biến sau đây:

- *Đào tạo từ đầu*

Để đào tạo một mạng lưới sâu từ đầu, phương pháp này đòi hỏi phải thu thập một tập dữ liệu lớn được gắn nhãn và thiết kế một kiến trúc mạng. Điều này tốt cho các ứng dụng mới hoặc các ứng dụng sẽ có số lượng các danh mục đầu ra lớn. Đây là một cách ít phổ biến vì với lượng dữ liệu và tốc độ học tập lớn, các mạng này thường mất vài ngày hoặc vài tuần để đào tạo.

- *Học chuyển giao*

Đây là phương pháp chủ yếu trong Deep Learning, là một quá trình liên quan đến việc tinh chỉnh một mô hình đã được sàng lọc trước. Nó bắt đầu với một mạng hiện có, sau đó cung cấp thêm dữ liệu mới có chứa các lớp mà trước đó chưa biết. Sau khi thực hiện một số điều chỉnh bổ sung và điều chỉnh dữ liệu cho phù hợp và chính xác hơn, từ đó nó có thể thực hiện một tác vụ mới, chẳng hạn như điều chỉnh phân loại theo đối tượng giúp cho thời gian tính toán giảm xuống đáng kể.

- *Đào tạo theo cách xếp chồng tự động*

Bộ dữ liệu huấn luyện được xếp chồng lên nhau một cách tự động để tạo thành một mạng sâu và tồn tại dạng tiềm ẩn, đầu ra của nó có thể làm đầu vào cho lớp hiện tại. Mỗi lớp được huấn luyện và sàng lọc để tăng độ chính xác cho đầu vào (và là mẫu đầu ra của lớp trước) và tiếp tục việc tự động xếp chồng một lớp khác.

Ngày nay, với sự phát triển mạnh mẽ của Công nghệ thông tin, trí tuệ nhân tạo càng trở nên phổ biến, vì thế Deep Learning cũng là xu hướng phát triển của thời đại nhờ tính ưu việt và độ chính xác của nó.

2.2. MÔI TRƯỜNG VÀ CÔNG CỤ SỬ DỤNG

2.2.1. Thư viện mã nguồn Tensorflow Object Detection

TensorFlow ™ là một thư viện phần mềm nguồn mở để tính toán số hiệu suất cao. Kiến trúc linh hoạt của nó cho phép dễ dàng triển khai tính toán trên nhiều nền tảng khác nhau (CPU, GPU, TPU) và từ máy tính để bàn đến các cụm máy chủ tới thiết bị di động. Được phát triển bởi các nhà nghiên cứu và kỹ sư từ nhóm Google Brain trong tổ chức AI của Google, nó hỗ trợ mạnh mẽ cho Machine Learing,Deep Learning và lõi tính toán linh hoạt được sử dụng trên nhiều lĩnh vực khoa học khác.

- *Các thành phần trong TensorFlow:*
  - Node
  - Tensor
  - Rank
  - Shape
  - Op
  - DType

2.2.2. Thuật toán SSD (Single Shot Object Detectors)

a. Giới thiệu

Để hiểu rõ hơn về SSD, hãy bắt đầu bằng cách giải thích tên của kiến trúc này đến từ đâu:

- Single Shot: Có nghĩa là các nhiệm vụ của định vị hóa và phân loại đối tượng được thực hiện trong một lần chuyển tiếp duy nhất của vật thể.

- MultiBox: Tên của một kỹ thuật hộp ràng buộc được phát triển bởi Szegedy etal.
- Detector: Là một bộ dò tìm đối tượng và phân loại các đối tượng được phát hiện.

SSD sử dụng các layer có kích thước giảm dần theo độ sâu để nhận dạng đối tượng. SSD có độ phân giải giảm đáng kể qua mỗi layer và sẽ bỏ sót những đối tượng có kích thước nhỏ ở những lớp có độ phân giải thấp, vì vậy trong thực tế hình ảnh đầu vào cần phải có độ phân giải cao.

Thuật toán SSD chạy khá nhanh, nhưng độ chính xác của nó không cao (không bằng region proposal).

b. MultiBox

Kỹ thuật hồi quy hộp giới hạn của SSD được lấykết quả từ công trình của Szegedy trên MultiBox, một phương pháp cho các đề xuất điều phối hộp giới hạn nhanh, nhạy cảm với lớp học. Quá trình xử lý được thực hiện trên MultiBox, mạng chuyển đổi kiểu khởi động được sử dụng.

Hàm tinh giảm của MultiBox cũng kết hợp hai thành phần quan trọng đã được đưa vào SSD:

- Confidence Loss.
- Location Loss.

Nếu không đi sâu vào toán học thì biểu thức cho hàm tinh giản, tức đo lường mức độ dự đoán của chúng ta "đúng" như thế nào:

multibox_loss = confidence_loss + alpha * location_loss

Thuật ngữ alpha giúp chúng ta cân bằng sự đóng góp của việc mất vị trí. Như thường lệ trong Deep Learning, mục đích là tìm các giá trị tham số tối ưu nhất làm giảm hàm tinh giảm, do đó đưa các dự đoán của chúng ta sẽ gần hơn với thực tiễn.

c. Thuật toán phát hiện vật thể SSD

Qua các mô hình được phân tích ở trên chúng ta có thuật toán minh họa như sau (thuật toán phát hiện vật thể trong video, sử dụng thư viện opencv-python==4.1.0.25):

```
import cv2
print('Project Topic : Car Counting and Speed Detection')
print('Research Internship on Machine learning using Images')
print('By Dang Thi My Nhan')
cascade_src = 'cars.xml'
video_src = 'video.avi'
cap = cv2.VideoCapture(video_src)
car_cascade = cv2.CascadeClassifier(cascade_src)
while True:
ret, img = cap.read()
if (type(img) == type(None)):
break
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cars = car_cascade.detectMultiScale(gray, 1.1, 2)
for (x,y,w,h) in cars:
cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,255),2)
cv2.imshow('video', img)
if cv2.waitKey(33) == 27:
break
cv2.destroyAllWindows()
```

2.2.3. Thuật toán Faster R-CNN

Faster R-CNN là một thuật toán để tìm kiếm vị trí của vật thể trong ảnh.Thuật toán này sẽ có đầu ra là những hình hộp, cùng với vật thể bên trong hộp đó là gì. Phiên bản đầu tiên của Faster R-CNN là R-CNN, với nguyên lý khá đơn giản.

Sử dụng các mạng huấn luyện sẵn để feed-forward các region proposals, sẽ tốn nhiều thời gian bởi với mỗi ảnh thuật toán selective search sẽ cho ra hàng nghìn region proposals.

Với thuật toán Faster R-CNN, chúng ta sử dụng một convolution filter trả ra 5 kết quả dự đoán: 4 giá trị là toạ độ của boundary box, và giá trị còn lại là xác suất xuất hiện đối tượng. Tổng quát hơn, ta có input là D feature map 8x8, output là 8x8x5, số convolution filter trong Faster R-CNN là 3x3xDx8.

Thử nghiệm một lần đối với ảnh gốc, thu được convolutional features của ảnh đó. Ví dụ với một hình ảnh có kích thước 600∗600∗3600∗600∗3, ta sẽ thu được convolutional features với kích thước 37∗37∗51237∗37∗512. Kích thước của features bị giảm nhỏ khoảng 16 lần 6003760037.

Dựa vào kích thước cùng vị trí của các region proposals đối với ảnh gốc, ta sẽ tính toán được vị trí của region proposal trong convolutional features.

Sử dụng giá trị convolutional faetures của region proposal, ta dự đoán được vị trí các đỉnh của bounding box cũng như vật thể nằm trong bounding box là gì.

Đối với Fast R-CNN, do chia sẻ tính toán giữa các region trong ảnh, tốc độ thực thi của thuật toán đã được giảm từ 120s mỗi ảnh xuống 2s. Phần tính toán gây ra nghẽn chính là phần đưa ra các region proposal đầu vào, chỉ có thể thực thi tuần tự trên CPU.Faster R-CNN giải quyết vấn đề này bằng cách sử dụng DNN để tính toán các region proposals này.

So với thuật toán Faster R-CNN thì thuật toán SSD chạy khá nhanh, nhưng có độ chính xác không cao, đặc biệt khi việc nhận dạng các đối

tượng có kích thước nhỏ. Trong khi đó Faster R-CNN cho kết quả có độ chính xác cao kể cả đối với hình ảnh đầu có độ phân giải thấp.

2.2.4. Lựa chọn mô hình Object Detectors

Trong luận văn này, chúng ta sẽ lựa chọn mô hình Object Detector cho bài toán.

Các thuật toán Object Detection đã phát triển trong một khoảng thời gian dài. Ý tưởng đầu tiên, đơn giản nhất là chúng ta sẽ sử dụng cửa sổ trượt.

Sliding windows

for window in windows

patch = get_patch(image, window)

results = detector(patch)

Để tăng tốc, chúng ta sẽ: 1. Giảm số lượng windows (R-CNN giảm còn khoảng 2000); 2.Giảm các phép tính trong việc tìm ROI (Fast R-CNN sử dụng feature map thay vì toàn bộ image patchs).

Fast R-CNN

feature_maps = process(image)

ROIs = region_proposal(feature_maps)

for ROI in ROIs

patch = roi_pooling(feature_maps, ROI)

results = detector2(patch)

Việc tìm region_proposal cũng tốn khá nhiều thời gian.Faster R-CNN sử dụng một convolution network thay thế cho region proposal ở bước này (làm giảm thời gian từ 2.3s xuống còn 0.3 giây). Faster R-CNN cũng giới thiệu 1 khái niệm là anchor giúp cải thiện độ chính xác và việc huấn luyện trở nên dễ dàng hơn.

R-FCN đưa ra một điều chỉnh nhỏ, là tiến hành tìm position và sensitive score map trên mỗi ROIS độc lập. Và tính trung bình xác suất xuất hiện đối tượng.

```
# R-FCN
feature_maps = process(image)
ROIs = region_proposal(feature_maps)
score_maps = compute_score_map(feature_maps)
for ROI in ROIs
  V = pool(score_maps, ROI)
class_scores = average(V)
  class_probabilities = softmax(class_scores)
```

R-FCN chạy khá nhanh, nhưng độ chính xác thì thấp hơn một chút so với Faster R-CNN.Thuật toán SSD (Single Shot Detector) được đề xuất để sử dụng 1 lần tính toán.

```
feature_maps = process(image)
results = detector3(feature_maps) # No more separate step for ROIs
```

Thuật toán SSD và YOLO đều thuộc nhóm single shot detectors. Cả hai đều sử dụng convolution layer để rút trích đặc trưng và một convolution filter để đưa quyết định. Cả hai đều dùng feature map có độ phân giải thấp (low resolution feature map) để dò tìm đối tượng => chỉ phát hiện được các đối tượng có kích thước lớn. Một cách tiếp cận là sử dụng các feature map có độ phân giải cao (higher resolution feature map). Nhưng độ chính xác sẽ giảm do thông tin đặc trưng của đối tượng quá hỗn loạn. FPN đưa ra ý tưởng sử dụng feature map trung gian merge giữa feature map high resolution và low resolution. Việc này giúp cho chúng ta vẫn giữ được thông tin đặc trưng hữu ích của đối tượng, đồng thời cũng giữ được thông tin của các đối tượng có kích thước nhỏ. Do đó, độ chính xác cũng tăng lên và phát hiện các đối tượng có các tỷ lệ khác nhau (different scale) tốt hơn.

Trong quá trình huấn luyện, chúng ta sẽ nhận ra 1 vấn đề rằng backgroup sẽ chiếm 1 phần rất lớn trong bức ảnh.Hoặc một đối tượng nào đó có số mẫu nhiều hơn so với các đối tượng khác.Thuật toán Focal loss được sinh ra để giải quyết vấn đề này.

2.3. PHÁT TRIỂN CHƯƠNG TRÌNH

Để đào tạo mô hình Machine Learning nhận dạng được phương tiện và tính tốc độ xe thì chúng ta cần thực hiện nhiều bước theo quy trình như sau:

[Hình 2.9] Các bước giải quyết bài toán.

2.3.1. Thu thập dữ liệu

Từ video đã có, ta dùng thư viện OpenCV để cắt video ra thành nhiều bức ảnh, trung bình với 1 video khoảng 40s ta được 50 bức hình:

covert_video_to_image.py

```
import cv2
path = "../Dang Nhan/Dataset_all/DNG33"
video = "v203161"
cap = cv2.VideoCapture(path + "/"+ video +".ts")
frameCount = cap.get(cv2.CAP_PROP_FRAME_COUNT)
i = 0
```

```
while True:
ret, image = cap.read()
if ret == False:
break
i += 1
if i % 5 == 0:
link = path + "/" + video + "/" + video + " " + str(i) + ".JPG"
cv2.imwrite(link, image)
cap.release()
```

Đối với đề tài này, dữ liệu được lấy từ các camera giao thông ở địa bàn thành phố Đà Nẵng và các nguồn khác từ internet.

	Dữ liệu huấn luyện	Dữ liệu kiểm thử
Size	Kích thước bất kì	Kích thước bất kì
Số lượng	2000	10

[Bảng 2.1] Test dữ liệu huấn luyện

2.3.2. Đánh nhãn dữ liệu

Với dữ liệu ảnh ta đã thu thập được ở trên, ta dùng phần mềm labelImage để gán nhãn cho dữ liệu.

Ta dùng phần mềm để vẽ khung ô tô cho tất cả các xe ô tô có trong ảnh, sau đó ta gán nhãn cho khung đó là ‘car’. Sau khi làm xong, ta lưu lại được file chú thích .xml.

2.3.3. Đào tạo mô hình

- Tải Tensorflow về máy tính

$ git clone https://github.com/tensorflow/models.git

- Cài đặt thư viện cần thiết

$ pip install tensorflow-gpu

- Từ Folder tensorflow/models/research xuất ra các file proto

$ protoc object_detection/protos/*.proto --python_out=.

- Đánh nhãn dữ liệu 2 lớp:

item {

id: 1

name: 'motocycle'

}

item {

id: 2

name: 'car'

}

Từ Folder models/object_detection chạy code training dữ liệu

$pythontrain.py–logostderr--train_dir=training/ pipline_config_path=training/ssd_mobilenet_v1_pets.config

2.3.4. Nhận dạng

Mục tiêu của bài toán là từ một video cụ thể thì phải nhận diện ra được các phương tiện xuất hiện trong video đó.

Cụ thể được trình bày như sau:

- Load video

cap = cv2.VideoCapture('test_video/vehicle_video.mp4')

- Lấy từng frame và xử lý nhận dạng

(ret, frame) = cap.read()

input_frame = frame

image_np_expanded = np.expand_dims(input_frame, axis=0)

(boxes, scores, classes, num) = sess.run([detection_boxes, detection_scores, detection_classes, num_detections], feed_dict={image_tensor: image_np_expanded})

2.3.5. Đếm phương tiện

Sau khi đã có mô hình để nhận dạng, ta bắt đầu đưa video rồi xác định vùng nhận diện khả dụng (ROI) trên video. Tiếp theo sẽ tách từng frame ra, nhận dạng xe trong từng frame, update liên tục vị trí của xe, nếu xe đã vào trong ROI thì tăng biến đếm lên. Cứ lặp lại như vậy cho đến khi kết thúc video thì xuất ra kết quả đã đếm được bao nhiêu xe.

- Gọi hàm để xử lý kết quả của việc nhận dạng

counter = vis_util.visualize_boxes_and_labels_on_image_array(

cap.get(1),input_frame,

np.squeeze(boxes),

np.squeeze(classes).astype(np.int32),

np.squeeze(scores),

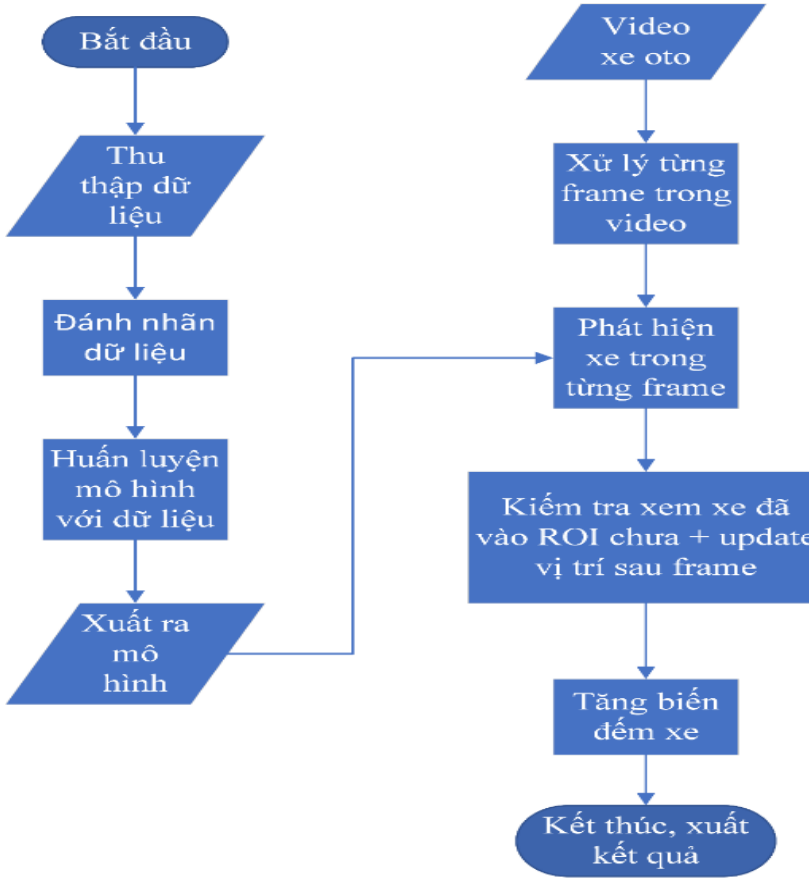
category_index,

use_normalized_coordinates=True,

line_thickness=4

)

- Quy trình nhận dạng và đếm xe:



[Hình 2.16] Quy trình nhận dạng và đếm xe.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. KẾT QUẢ THỰC NGHIỆM

Thực nghiệm trên một video mẫu được trích xuất từ camera giao thông tại đoạn đường gần trường Trường Mầm Non Tiên Sa (106 Quang Trung, phường Thạch Thang, quận Hải Châu, TP. Đà Nẵng) vào ngày 01/4/2019:

[Hình 3.1] Kết quả nhận diện và đếm xem qua video giao thông.

3.1.1. Nhận diện xe chuyển động

Sau quá trình huấn luyện ta thu được một mô hình có thể nhận dạng xe, đếm số lượng xe. Để đánh giá khả năng nhận dạng xe của mô hình, chạy thử với mô hình nhận dạng 60 giây đầu tiên của video trích xuất từ camera giám sát có chứa xe ở nhiều góc, kích thước lớn nhỏ khác nhau

Số lượng ô tô xuất hiện trong video ghi được không nhiều, do đó tác giả tiến hành thử nghiệm nhận dạng trên cả 2 loại phương tiện ô tô vàxe máy. Kết quả nhận diện tương đối chính xác, tuy nhiên có một số hình ảnh ô tô được nhận diện chưa chính xác do những mẫu ô tô đó không có trong tập huấn luyện, số lượng mẫu ô tô được sử dụng để huấn luyện chưa nhiều, mẫu ô tô chưa đa dạng.

3.1.2. Đếm số lượng xe

Kết quả đếm số lượng ô tô và xe máy từ camera giao thông nói trên là tương đối chính xác đối với những dữ liệu mẫu đã được huấn luyện, một số phương tiện không đếm được do không có trong tập dữ liệu huấn luyện hoặc do các tác động khách quan như bị vật cản che khuất, xe chạy sát

nhau,... Có trường hợp đếm nhầm do các vật thể có hình dạng giống ô tô hoặc xe máy.

STT	Số xe máy thực tế	Số xe hơi thực tế	Số xe máy đếm được	Số xe hơi đếm được	Độ lệch (Thực tế - Đếm được)
0 – 60	3	2	2	2	80,00%
61 – 200	7	2	5	2	77,78%
201 – 400	7	3	6	2	80,00%
401 – 600	6	3	4	1	55,56%
601 – 800	9	3	3	2	41,67%
801 – 1000	5	2	3	2	71,43%
1001 - 1200	6	1	5	1	85,71%
1200 - 1400	7	3	3	2	50,00%
1400 - 1600	11	3	4	2	42,86%

[Bảng 3.1] Số liệu nhận dạng và đếm xe

3.2. ĐÁNH GIÁ KẾT QUẢ

3.2.1. Nhận diện xe chuyển động

Kết quả nhận dạng xe của mô hình là khá cao khi mà các xe lưu thông ở đường phố thì có rất nhiều sự khác biệt nhau về màu sắc, kích thước, số lượng người ngồi trên xe hay loại hàng hóa chở trên xe, và còn bị ảnh hưởng bởi nhiều yếu tố khác như bị che khuất, bóng cây, thời tiết,…

Qua thực nghiệm thì nhận thấy rằng mô hình cho kết quả nhận dạng cao nhất khi mà các xe không bị che khuất, và có kích thước vừa phải.

Đối với một số trường hợp xe không nhận dạng được thì chủ yếu là do bị che khuất một phần, kích thước hình ảnh xe quá nhỏ và xe chưa được huấn luyện.

3.2.2. Đếm số lượng phương tiện

Độ chính xác của hệ thống đếm phương tiện là khá cao.Kết quả triển khai bài toán này như ta thấy ở trên vẫn có nhiều khả quan khi ứng dụng vào bài toán yêu cầu độ chính xác không quá lớn, có thể đưa ra con số chấp nhận được. Từ hệ thống đơn giản này, ta có thể cải tiến để nâng cao khả năng nhận dạng xe cũng như độ chính xác của bài toán phức tạp.

Kết quả của bài toán phụ thuộc rất nhiều vào tình hình giao thông, thời tiết, địa điểm thực hiện và nhiều yếu tố khác.Mật độ lưu thông của các phương tiện giao thông càng lớn thì độ chính xác cũng giảm dần. Cũng tương tự nếu thời tiết diễn biến xấu hay địa điểm đếm xe có nhiều vật cản che khuất hay có nhiều bóng cây. Bên cạnh đó thuật toán chưa ổn định cũng là một yếu tố ảnh hưởng đến kết quả.

PHẦN III: TỔNG KẾT

1. Kết luận

Trong nghiên cứu này, tôi đã đề cập đến các nghiên cứu về nhận dạng và phát hiện đối tượng trong video trước đây đã được thử nghiệm, đánh giá, tìm ra điểm chung giữa các nghiên cứu. Từ đó đề ra giải pháp tối ưu hơn bằng việc lựa chọn phương pháp Deep Learning để thực hiện việc nhận dạng và đếm xe ôtô thông qua video.

Sau quá trình triển khai đề tài, về cơ bản hệ thống đã có thể nhận dạng và đếm xe với độ chính xác khá cao. Tuy nhiên, vẫn có một trường hợp chưa nhận dạng được. Sự thiếu chính xác này xuất phát từ nhiều nguyên nhân khác nhau như do thiếu dữ liệu đào tạo nên mô hình nhận dạng có hiệu suất không cao, thuật toán tính vận tốc xe vẫn chưa tối ưu, hay do các nguyên nhân khách quan khác như vật cản che khuất, thời tiết,...

Hệ thống này sẽ đạt hiệu suất cao nhất khi nhận diện xe trên đường một chiều, làn đường rộng và xe mật độ xe không quá cao.

2. Hướng phát triển

Để cải thiện độ chính xác, ta sẽ tiến hành thu thập thêm nhiều dữ liệu hơn, đa dạng hơn. Từ đó sẽ giúp mô hình nhận dạng xe chính xác hơn. Ngoài ra để nhận diện và tính toán vận tốc xe hiệu quả, ta cũng cần cải thiện thuật toán nhận dạng và tính toán.

Sau khi có một hệ thống nhận diện xe có hiệu suất ổn định, đếm chính xác lưu lượng xe ô tô tham gia giao thông trên đường ta sẽ ứng dụng nó vào những bài toán khác lớn hơn trong thực tế để giải quyết các vấn đề về tắc nghẽn giao thông.