HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐẶNG KIM THÀNH

XÂY DỰNG ỨNG DỤNG NHẬN DIỆN BIỂN KIỂM SOÁT PHƯƠNG TIỆN GIAO THÔNG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐẶNG KIM THÀNH

XÂY DỰNG ỨNG DỤNG NHẬN DIỆN BIỂN KIỂM SOÁT PHƯƠNG TIỆN GIAO THÔNG

Chuyên ngành : **KỸ THUẬT VIỄN THÔNG**

Mã số : **8.52.02.08**

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN NGỌC MINH

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Đặng Kim Thành

MỤC LỤC

DANH MỤC BẢNG BIỂU, HÌNH ẢNH

DANH MỤC TỪ VIẾT TẮT

AI: Artifical Intelligence	Trí tuệ nhân tạo
ANN: Artificial Neural Network	Mạng nơ-ron nhân tạo
CAP: Credit assignment path	Đường gán kế thừa
CNN: Convolutional Neural Network	Mạng nơ-ron tích chập
DL: Deep learning	Học sâu
DN: Development Network	Mạng phát triển
HOG: Histogram of oriented gradient	Biểu đồ định hướng
ML: Machine learning	Học máy
RCNN: Region Convolutional Neural Network	Mạng nơ-ron tích chập
SSD: Single shot object detectors	Phát hiện đối tượng đơn

MỞ ĐẦU

Mục tiêu

- Xây dựng ứng dụng nhận diện biển kiểm soát phương tiện giao thông

Nội dung

- Nghiên cứu thực trạng
- Nghiên cứu tổng quan
- Lựa chọn giải pháp tiến hành
- Nghiên cứu các vấn đề lý thuyết
- Xây dựng hệ thống phần mềm thu thập dữ liệu, xử lý hình ảnh biển kiểm soát phương tiện giao thông

Đề tài: "Xây dựng ứng dụng nhận diện biển kiểm soát phương tiện giao thông" sẽ trình bày với các phần như sau:

- Chương 1: Tổng quan về nhận dạng biển kiểm soát phương tiện giao thông
- Chương 2: Các vấn đề cơ bản về nhận dạng, xử lý ảnh
- Chương 3: Xây dựng thuật toán nhận diện biển kiêm soát phương tiện giao thông
- Chương 4: Kết quả và bàn luận

CHƯƠNG 1. TỔNG QUAN VỀ NHẬN DẠNG BIỂN KIỂM SOÁT PHƯƠNG TIỆN GIAO THÔNG

Trong chương này, tác giả xin được trình bày tổng quan về nhận dạng biển kiểm soát phương tiện giao thông, vai trò cũng như thực trạng về nhận dạng biển kiểm soát giao thông ở Việt Nam và trên thế giới

1.1. Vai trò nhận dạng biển kiểm soát phương tiện giao thông trong công tác điều tra, khám phá các vụ việc mang tính hình sự của lực lượng Kỹ thuật hình sự

Trong quá trình phát triển của con người, những cuộc cách mạng về công nghệ đóng một vai trò rất quan trọng, chúng làm thay đổi từng ngày từng giờ cuộc sống của con người, theo hướng hiện đại hơn. Đi đôi với quá trình phát triển của con người, tình hình tội phạm và các vụ việc mang tính hình sự đang ngày càng gia tăng, đồng thời với sự phát triển không ngừng của các ngành Khoa học kỹ thuật dẫn đến phương thức thủ đoạn phạm tội đang ngày càng tinh vi, khó nắm bắt hơn và hầu hết đều có liên quan đến các phương tiên giao thông.

Ở các nước trên thế giới và nước ta hiện nay, việc nhận diện biển kiểm soát phương tiện giao thông đã được triển khai phổ biến trong lĩnh vực an ninh. Cụ thể ở Việt Nam, năm 2012 là năm an toàn giao thông (ATGT). Nhiều công trình nghiên cứu của các bộ, ngành đang được thực hiện để kiềm chế tai nạn giao thông (TNGT), giảm ùn tắc giao thông. Trong đó, đề tài "Ứng dụng phần mềm nhận dạng biển số trong quản lý phương tiện và phát hiện vi phạm giao thông bằng hình ảnh" của Cục Quản lý khoa học - công nghệ và môi trường (Tổng cục Hậu cần kỹ thuật - Bộ công an), phối hợp Công ty cổ phần Biển Bạc là một trong những đề tài đang được Áp dụng, triển khai tại một số tỉnh, thành phố trong cả nước ...

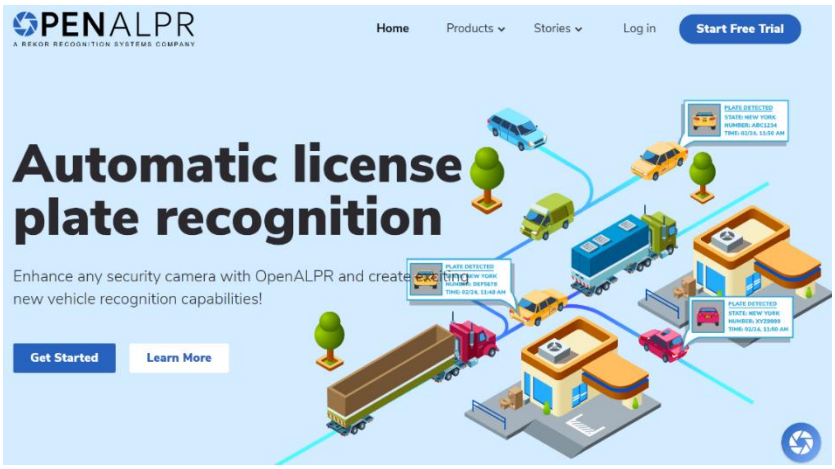
Vấn đề nhược điểm trong quá trình sử dụng phần mềm này là cần sự hỗ trợ của một hệ thống camera chuyên biệt, có tốc độ chụp hình cao và độ phân giải lớn. Trong khi đó việc sử dụng dữ liệu từ camera an ninh thông thường để nhận diện biển số lại

chưa được hiệu quả. Chính vì vậy, tôi chọn đề tài "Xây dựng ứng dụng nhận diện biển kiểm soát phương tiện giao thông" hướng đến cụ thể trong việc sử dụng dữ liệu hình ảnh từ các camera an ninh thông thường

1.2. Thực trạng về nhận dạng biển kiểm soát phương tiện giao thông ở Việt Nam và trên thế giới

Trên thế giới nói chung hay Việt Nam nói riêng, bài toán nhận diện biển kiểm soát phương tiện giao thông vẫn là một bài toán hết sức được quan tâm. Kết quả nghiên cứu bài toán là các giải pháp cho các công nghệ tự động, giải pháp quản lý bãi xe, quản lý an ninh, theo dõi tội phạm, ...

Nhìn chung các giải pháp hiện nay đang rất phát triển song vẫn còn nhiều hạn chế. Những giải pháp nổi tiếng trên thế giới như ALPR, Vino đều đáp ứng được hầu hết các nhu cầu cơ bản của các bài toán. Nhược điểm còn tồn tại trong các giải pháp này là chưa hỗ trợ hết tất cả các loại biển số trên thế giới.



Hình 1.1: Giải pháp OpenALPR trong nhận diện biển kiểm soát

Tại Việt Nam thị trường giải pháp nhận diện biển kiểm soát cũng rất phong phú, đi đầu trong lĩnh vực này có thể kể đến như Silver Sea, DNC Tech. Tất cả các giải pháp này đều có điểm chung là xử lý chuyên về biển số Việt Nam nên có tốc độ xử lý rất nhanh và độ chính xác rất cao đối với biển kiểm soát Việt Nam

CHƯƠNG 2. CÁC VẤN ĐỀ CƠ BẢN VỀ XỬ LÝ, NHẬN DẠNG ẢNH

Trong chương này tác giả sẽ giới thiệu tổng quan về các vấn đề trong xử lý ảnh; nhận dạng ảnh; các phương pháp, thuật toán xử lý ảnh; Ứng dụng của trí tuệ nhân tạo trong xử lý ảnh;

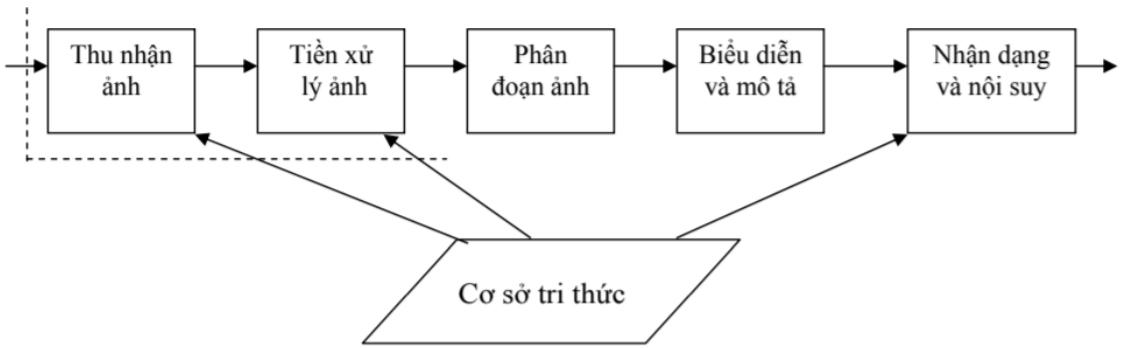
2.1. Tổng quan về xử lý ảnh

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xửlý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thứ hai, các công cụ toán như Đại số tuyến tính, xác suất, thống kê. Một số kiến thứ cần thiết như trí tuệ nhân tao, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh sô thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo

như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.



Hình 2.1: Các bước cơ bản trong xử lý ảnh

Sơ đồ này bao gồm:

a) Phần thu nhận ảnh

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hoá (như loại CCD – Change Coupled Device) là loại photodiot tạo cường độ sáng tại mỗi điểm ảnh. Camera thường dùng là loại quét dòng ; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh)

b) Phần tiền xử lý ảnh

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

c) Phần phân đoạn ảnh

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về địa chỉ hoặc tên người

thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

d) Biểu diễn ảnh

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lận cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

e) Nhận dạng và nội suy

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại. Có nhiều cách phân loai ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản: Nhận dạng theo tham số và nhận dạng theo cấu trúc. Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người…

f) Cơ sở tri thức

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiễu. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý

ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy. Trong tài liệu, chương 6 về nhận dạng ảnh có nêu một vài ví dụ về cách sử dụng các cơ sở tri thức đó

g) Biểu diễn ảnh

Sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (Image Features) như: biên ảnh (Boundary), vùng ảnh (Region). Một số phương pháp biểu diễn thường dùng:

- Biểu diễn bằng mã chạy (Run-Length Code)
- Biểu diễn bằng mã xích (Chaine -Code)
- Biểu diễn bằng mã tứ phân (Quad-Tree Code)

2.2. Lý thuyết nhận dạng ảnh, nhận dạng đối tượng

2.2.1. Tổng quan về bài toán nhận dạng

Nhận dạng hay nhận biết một đối tượng là khả năng tự nhiên của con người cũng như các loài vật. Theo một cách bản năng nhất mọi loài vật qua cảm nhận từ các cơ quan cảm giác như: mắt, mũi, miệng, tay, ... bằng một hành động nhìn, nghe, ngửi, ... có thể cảm nhận ngay được đối tượng đang tiếp xúc với mình là cái gì, quen hay lạ. Chính vì vậy có thể nói răng cơ thể mỗi loài vật chính là một hệ thống nhận dạng tối ưu nhất.

Với sự phát triển của khoa học công nghệ nhất là khoa học về robot thì càng ngày mong ước tạo ra một hệ thống nhận dạng máy học có khả năng tương tự thậm chí là vượt trội hơn hệ thống nhận dạng sinh học là một khát khao của các nhà khoa học

Một hệ thống nhận dạng đối tượng là hệ thống nhận vào một ảnh hoặc một đoạn video (chuỗi các ảnh). Qua xử lý tính toán, hệ thống xác định được vị trí đối tượng đang có trong ảnh (nếu có) và xác định được đó là đối tượng nào trong số những đối tượng mà hệ thống đã biết (qua quá trình học) hay là đối tượng mới. Yêu cầu đặt ra với các hệ thống nhận dạng đối tượng là độ chính xác cao vì vậy hệ thống đòi hỏ phải có các đặc trưng tốt. Hệ thống phải biết chọn các đặc trưng như thế nào để có thể biểu diễn tốt được thông tin đối tượng cần nhận dạng. Đồng thời, đặc trưng phải được tính toán nhanh để không làm chậm công việc nhận dạng. Thêm vào đó, hệ thống phải có phương pháp học hiệu quả, có khả năng nhận dạng tốt các mẫu mới chứ không chỉ làm tốt trên các mẫu đã học.

2.2.2. Các khó khăn của công việc nhận dạng

Đối với con người thì việc nhận dạng các đối tượng trong ảnh là việc không phải là phức tạp; tuy nhiên đối với một hệ thống nhân tạo thì việc nhận ra một đối tượng từ một ảnh đòi hỏi phải giải quyết được rất nhiều vấn đề. Chính vì thế vấn đề này vẫn đang được nhiều nhóm trên thế giới quan tâm nghiên cứu, Khó khăn của việc nhận dạng có thể kể ra như sau:

a) Tư thế, góc chụp: Ảnh chụp khuôn mặt có thể thay đổi rất nhiều bởi vì góc chụp giữa camera và khuôn mặt. Chẳng hạn như chụp thẳng, chụp xéo bên trái hoặc xéo bên phải. Với các tư thế khác nhau, các thành phần trên khuôn mặt như mắt mũi, miệng có thể bị che khuất một phần hoặc thậm chí là che khuất hết

Hình 2.2: Minh họa sự thay đổi góc chụp

b) Sự xuất hiện hoặc thiếu một số thành phần: Các thành phần biểu tả một đối tượng có thể xuất hiện hoặc không trong ảnh làm cho bài toán nhận dạng càng trở nên khó hơn nhiều

Hình 2.3: Minh họa sự thiếu thành phần

c) Sự biến dạng của đối tượng: Biến dạng của đối tượng có thể làm ảnh hưởng đáng kể đến các thông số của đối tượng đó. Chẳng hạn cùng một khuôn mặt một người, nhưng có thể sẽ rất khác khi họ cười hoặc sợ hãi.

Hình 2.4: Minh họa sự biến dạng

d) Sự che khuất: Đối tượng có thể bị che khuất bởi các đối tượng khác

Hình 2.5: Minh họa sự che khuất

e) Sự phức tạp của hình nền: Hình nền phức tạp sẽ khiến việc nhận dạng trở nên khó khăn.

Hình 2.6: Minh họa hình nền phức tạp

f) Điều khiển của ảnh: Ảnh được chụp trong các điều kiện khác nhau về chiều sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại, ...) ảnh hưởng rất nhiều đến chất lượng ảnh.

Hình 2.7: Minh họa sự thay đổi độ sáng

2.2.3. Các ứng dụng nhận dạng đối tượng

Bài toán nhận dạng đối tượng có thể được ứng dụng rộng rãi trong nhiều ứng dụng thực tế khác nhau. Đó chính là lý do mà bài toán này hấp dẫn rất nhiều nhóm nghiên cứu trong thời gian dài. Các ứng dụng liên quan đến nhận dạng đốit ượng có thể kể đến như:

- Hệ thống phát hiện tội phạm: Camera được đặt tại một số điểm công cộng như siêu thị, nhà sách, trạm xe buýt, ... Khi phát hiện được sự xuất hiện của các đối tượng là tội phạm, hệ thống sẽ gởi thông điệp về cho trung tâm xử lý.
- Hệ thống theo dõi nhân sự trong một đơn vị: Giám sát giờ ra, vào của từng nhân viên
- Hệ thống giao tiếp người máy
- Hệ thống tìm kiếm hình ảnh, video
- Hệ thông báo mật dựa trên sinh trắc học
- Nhận dạng chữ in
- Công nghệ quản lý biển số xe

2.2.4. Tổng quan kiến trúc của một hệ thống nhận dạng đối tượng

Một hệ thống nhận dạng đối tượng thông thường được xử lý qua 4 bước sau:

- Thu nhận tín hiệu (hình ảnh, video) và tiền xử lý
- Trích chọn đặc trưng
- Phát hiện đối tượng
- Phân lớp đối tượng

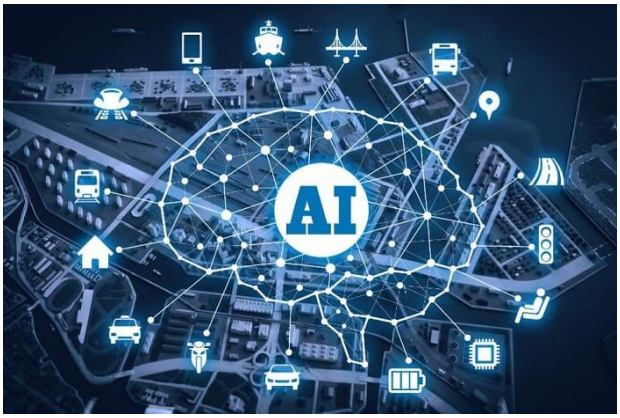
Ảnh đầu vào được thu nhận và tiền xử lý sau đó các ảnh được qua công đoạn tiền xử lý nhằm tăng độ chính xác cho hệ thống, Các ảnh sau đó được trích chọn đặc trưng để tạo ra các vector đặc trưng trong bước rút trích đặc trưng. Những vecto đặc trưng này sẽ là dữ liệu đầu vào cho một mô hình được huấn luyện trước. Phát hiện đối tượng: dò tìm và định vị những đối tượng xuất hiện trong ảnh. Những đối tượng thu được qua phát hiện sẽ tiếp tục được phân lớp thành các lớp riêng biệt để nhận dạng.

2.3. Ứng dụng của công nghệ trí tuệ nhân tạo trong xử lý ảnh

2.3.1. Tổng quan về công nghệ trí tuệ nhân tạo

Trong khoa học máy tính, trí tuệ nhân tạo hay AI (tiếng Anh: Artificial Intelligence), đôi khi được gọi là trí thông minh nhân tạo, là trí thông minh được thể hiện bằng máy móc, trái ngược với trí thông minh tự nhiên được con người thể hiện. Thông thường, thuật ngữ trí tuệ nhân tạo thường được sử dụng để mô tả các máy móc (hoặc máy tính) bắt chước các chức năng nhận thức mà con người liên kết với tâm trí con người, như học tập và giải quyết vấn đề. Khi máy móc ngày càng tăng khả năng, các nhiệm vụ được coi là cần trí thông minh thường bị loại bỏ khỏi định nghĩa về AI, một hiện tượng được gọi là hiệu ứng AI. Một câu châm ngôn trong Định lý của Tesler nói rằng AI là bất cứ điều gì chưa được thực hiện. Ví dụ, nhận dạng ký tự quang học thường bị loại trừ khỏi những thứ được coi là AI, đã trở thành một công nghệ thông thường. Khả năng máy hiện đại thường được phân loại như AI bao gồm thành công hiểu lời nói của con người,cạnh tranh ở mức cao nhất trong trò chơi chiến lược (chẳng

hạn như cờ vua và Go), xe hoạt động độc lập, định tuyến thông minh trong mạng phân phối nội dung, và mô phỏng quân sự.



Hình 2.8: Trí tuệ nhân tạo đã trở thành phần thiết yếu của công nghệ

Trí tuệ nhân tạo có thể được phân thành ba loại hệ thống khác nhau: trí tuệ nhân tạo phân tích, lấy cảm hứng từ con người và nhân tạo. AI phân tích chỉ có các đặc điểm phù hợp với trí tuệ nhận thức; tạo ra một đại diện nhận thức về thế giới và sử dụng học tập dựa trên kinh nghiệm trong quá khứ để thông báo các quyết định trong tương lai. AI lấy cảm hứng từ con người có các yếu tố từ trí tuệ nhận thức và cảm xúc; hiểu cảm xúc của con người, ngoài các yếu tố nhận thức và xem xét chúng trong việc ra quyết định. AI nhân cách hóa cho thấy các đặc điểm của tất cả các loại năng lực (nghĩa là trí tuệ nhận thức, cảm xúc và xã hội), có khả năng tự ý thức và tự nhận thức được trong các tương tác.

Trí tuệ nhân tạo được thành lập như một môn học thuật vào năm 1956, và trong những năm sau đó đã trải qua nhiều làn sóng lạc quan sau đó là sự thất vọng và mất kinh phí (được gọi là mùa đông AI), tiếp theo là cách tiếp cận mới, thành công và tài trợ mới. Trong phần lớn lịch sử của mình, nghiên cứu AI đã được chia thành các trường con thường không liên lạc được với nhau. Các trường con này dựa trên các cân nhắc kỹ thuật, chẳng hạn như các mục tiêu cụ thể (ví dụ: robot học hoặc học máy), việc sử dụng các công cụ cụ thể (logic hoặc mạng lưới thần kinh nhân tạo) hoặc sự khác biệt triết học sâu sắc. Các ngành con cũng được dựa trên các yếu tố xã hội (các tổ chức cụ thể hoặc công việc của các nhà nghiên cứu cụ thể).

Lĩnh vực này được thành lập dựa trên tuyên bố rằng trí thông minh của con người có thể được mô tả chính xác đến mức một cỗ máy có thể được chế tạo để mô phỏng nó.Điều này làm dấy lên những tranh luận triết học về bản chất của tâm trí và đạo đức khi tạo ra những sinh vật nhân tạo có trí thông minh giống con người, đó là những vấn đề đã được thần thoại, viễn tưởng và triết học từ thời cổ đại đề cập tới. Một số người cũng coi AI là mối nguy hiểm cho nhân loại nếu tiến triển của nó không suy giảm. Những người khác tin rằng AI, không giống như các cuộc cách mạng công nghệ trước đây, sẽ tạo ra nguy cơ thất nghiệp hàng loạt.

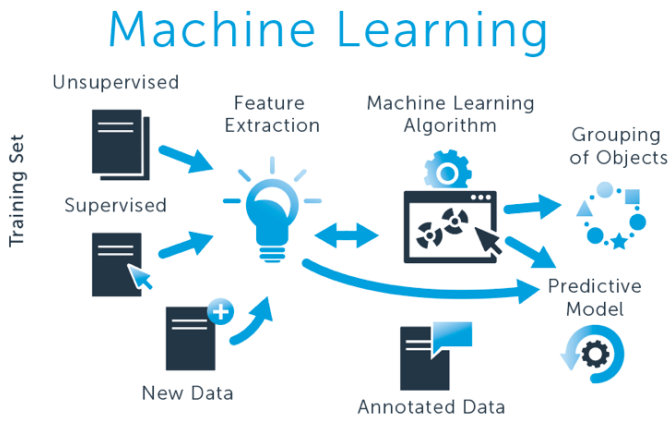
Hình 2.9: Nhờ trí tuệ nhân tạo, người máy đang giúp ích cho con người

Trong thế kỷ 21, các kỹ thuật AI đã trải qua sự hồi sinh sau những tiến bộ đồng thời về sức mạnh máy tính, dữ liệu lớn và hiểu biết lý thuyết; và kỹ thuật AI đã trở thành một phần thiết yếu của ngành công nghệ, giúp giải quyết nhiều vấn đề thách thức trong khoa học máy tính, công nghệ phần mềm và nghiên cứu vận hành.

2.3.2. Giới thiệu về học máy (Machine Learning)

Học máy (tiếng Anh: Machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ.

Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được. Học máy có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA (Deoxyribonucleic Acid), nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt.



Hình 2.10: Machine Learning – Một lĩnh vực của trí tuệ nhân tạo

Dưới góc nhìn của trí tuệ nhân tạo, động lực chính học máy bởi là nhu cầu thu nhận tri thức.Thật vậy, trong nhiều trường hợp ta cần kiến thức chuyên gia là khan hiếm (không đủ chuyên gia ngồi phân loại lừa đảo thẻ tín dụng của tất cả giao dịch hàng ngày) hoặc chậm vì một số nhiệm vụ cần đưa ra quyết định nhanh chóng dựa trên xử lý dữ liệu khổng lồ (trong mua bán chứng khoán phải quyết định trong vài khoảng khắc của giây chẳng hạn) và thiếu ổn định thì buộc phải cần đến máy tính. Ngoài ra, đại đa số dữ liệu sinh ra ngày nay chỉ phù hợp cho máy đọc tiềm tàng nguồn kiến thức quan trọng. Máy học nghiên cứu cách thức để mô hình hóa bài toán cho phép máy tính tự động hiểu, xử lý và học từ dữ liệu để thực thi nhiệm vụ được giao cũng như cách đánh giá giúp tăng tính hiệu quả.

Một số hệ thống học máy nỗ lực loại bỏ nhu cầu trực giác của con người trong việc phân tích dữ liệu, trong khi các hệ thống khác hướng đến việc tăng sự cộng tác giữa người và máy. Không thể loại bỏ hoàn toàn tác động của con người vì các nhà thiết kế hệ thống phải chỉ định cách biểu diễn của dữ liệu và những cơ chế nào sẽ được dùng để tìm kiếm các đặc tính của dữ liệu. Học máy có thể được xem là một nỗ lực để tự động hóa một số phần của phương pháp khoa học. Một số nhà nghiên cứu học máy tạo ra các phương pháp bên trong các khuôn khổ của thống kê Bayes.

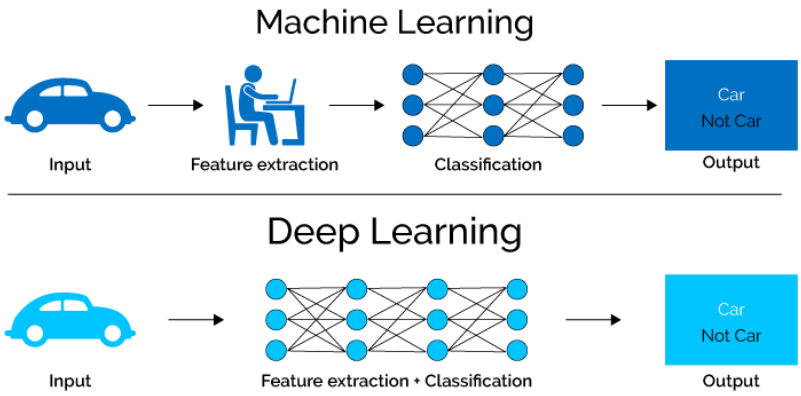
2.3.3. Giới thiệu về học sâu (Deep Learning)

Học sâu (tiếng Anh: Deep learning) là một chi của ngành máy học dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Học sâu là một phần của một họ các phương pháp học máy rộng hơn dựa trên đại diện học của dữ liệu. Một quan sát (ví dụ như, một hình ảnh) có thể được biểu diễn bằng nhiều cách như một vector của các giá trị cường độ cho mỗi điểm ảnh, hoặc một cách trừu tượng hơn như là một tập hợp các cạnh, các khu vực hình dạng cụ thể, vv. Một vài đại diện làm khiến việc học các nhiệm vụ dễ dàng hơn (ví dụ, nhận dạng khuôn mặt hoặc biểu hiện cảm xúc trên khuôn mặt) từ các ví dụ. Một trong những hứa hẹn của học sâu là thay thế các tính năng thủ công bằng các thuật toán hiệu quả đối với học không có giám sát hoặc nửa giám sát và tính năng phân cấp.

Các thuật toán học sâu tương phản với các thuật toán học nông bởi số biến đổi được tham số hóa một tín hiệu gặp phải khi nó lan truyền từ các lớp đầu vào đến lớp đầu ra, nơi một biến đổi được tham số hóa là một đơn vị xử lý có các thông số có thể huấn luyện được, chẳng hạn như trọng số và ngưỡng. Một chuỗi các biến đổi từ đầu vào đến đầu ra là một đường gán kế thừa (CAP- credit assignment path). CAP mô tả các kết nối quan hệ nhân quả tiềm năng giữa đầu vào và đầu ra và có thể thay đổi chiều dài. Đối với một mạng neuron nuôi tiến (feedforward), độ sâu của CAP, và do đó độ sâu của mạng đó, là số lượng các lớp ẩn cộng 1 (lớp đầu ra cũng là tham số

hóa). Đối với mạng neuron tái phát, trong đó một tín hiệu có thể truyền thông qua một lớp nhiều hơn một lần, CAP có khả năng không bị giới hạn chiều dài. Không có sự thống nhất chung về ngưỡng của độ sâu chia học cạn với học sâu, nhưng hầu hết các nhà nghiên cứu trong lĩnh vực đồng ý rằng học sâu có nhiều lớp phi tuyến để là học rất sâu.



Hình 2.11: Deep Learning – một phạm trù của Machine learning

Các thuật toán học sâu dựa trên các đại diện phân phối. Giả định tiềm ẩn đằng sau các đại diện phân phối là các dữ liệu được quan sát là được tạo ra bởi sự tương tác của các yếu tố được tổ chức theo lớp. Học sâu thêm giả định rằng các lớp của các yếu tố này tương ứng với các mức độ trừu tượng hay theo thành phần. Các con số khác nhau của các lớp và kích thước của lớp có thể được sử dụng để quy định các lượng trừu tượng khác.

Học sâu khai thác ý tưởng thứ bậc các yếu tố giải thích này ở cấp cao hơn, những khái niệm trừu tượng hơn được học từ các cấp độ thấp hơn. Những kiến trúc này thường được xây dựng với một phương pháp lớp chồng lớp tham lam. Học sâu giúp để tháo gỡ những khái niệm trừu tượng này và chọn ra những đặc điểm cần thiết cho việc học. Đối với các nhiệm vụ học có giám sát, các phương pháp học sâu sẽ tránh kỹ thuật đặc điểm (feature engineering), bằng cách dịch các dữ liệu vào các đại diện trung gian nhỏ gọn giống như các thành phần chính, và lấy được các cấu trúc lớp mà loại bỏ sự thừa thải trong đại diện. Rất nhiều các thuật toán học sâu được áp dụng cho các nhiệm vụ học không có giám sát. Đây là một lợi ích quan trọng bởi vì dữ liệu

không dán nhãn (chưa phân loại) thường phong phú hơn các dữ liệu dán nhãn. Một ví dụ của một cấu trúc sâu có thể được đào tạo theo cách không có giám sát là một mạng lưới tin sâu (deep belief network).

Trong năm 1991 những mạng neuron như vậy được sử dụng để nhận diện chữ số viết tay 2-D cách ly, nhận dạng đối tượng 3-D được thực hiện bằng cách kết hợp các hình ảnh 2-D với một mô hình đối tượng 3-D thủ công. Juyang Weng và các cộng sự đề xuất rằng một bộ não người không sử dụng một mô hình đối tượng 3-D nguyên khối, và vào năm 1992, họ xuất bản Cresceptron, một phương pháp để thực hiện nhận dạng đối tượng 3-D trực tiếp từ các hậu trường lộn xộn. Cresceptron là một ghép tầng của các lớp tương tự như Neocognitron. Nhưng trong khi Neocognitron yêu cầu một lập trình viên con người can thiệp, Cresceptron sẽ tự động học được một số đặc điểm không có giám sát trong mỗi lớp, nơi mà mỗi đặc điểm được đại diện bởi một nhân tích chập. Cresceptron cũng phân đoạn từng đối tượng học được từ một cảnh nền lộn xộn thông qua việc phân tích ngược mạng đó. Thăm dò max, bây giờ thường được thông qua bởi các mạng neuron sâu (ví dụ: các kiểm tra ImageNet), lần đầu tiên sử dụng trong Cresceptron để giảm độ phân giải vị trí bởi của một hệ số (2x2) đến 1 thông qua việc ghép tầng tổng quát hóa tốt hơn. Mặc dù có những lợi thế như thế, các mô hình đơn giản hơn sử dụng nhiệm vụ cụ thể có đặc điểm thủ công như bộ Gabor và các máy hỗ trợ vector (SVM-support vector machines) đã là lựa chọn phổ biến trong thập niên 1990 và thập niên 2000, bởi vì chi phí tính toán bởi các ANN và vì thiếu sự hiểu biết về cách thức bộ não tự quản các kết nối mạng sinh học của nó.

Một số phương pháp học sâu thành công nhất là mạng neuron nhân tạo. Mạng neuron nhân tạo được lấy cảm hứng từ các mô hình sinh học năm 1959 được đề xuất bởi người đoạt giải Nobel David H. Hubel & Torsten Wiesel, 2 người đã tìm thấy hai loại tế bào trong vỏ não thị giác chính: các tế bào đơn giản vàcác tế bào phức tạp. Nhiều mạng neuron nhân tạo có thể được xem như là các mô hình ghép tầng của các tế bào loại lấy cảm hứng từ những quan sát sinh học. Neocognitron của Fukushima giới thiệu các mạng neuron tích chập được đào tạo một phần bởi học không có giám

sát với các đặc điểm được con người hướng dẫn trong mặt phẳng thần kinh. Yann LeCun...(1989) áp dụng truyền ngược có giám sát cho các kiến trúc như vậy. Weng... (1992) công bố các mạng neuron tích chập Cresceptron để nhậ dạng các đối tượng 3-D từ các hình ảnh có hậu trường lộn xộn và phân khúc của các đối tượng từ hình ảnh đó. Một nhu cầu rõ ràng để nhận dạng các đối tượng 3-D nói chung là ít nhất là thay đổi tính bất biến và khả năng chịu biến dạng. Thăm dò Max (Max-pooling) xuất hiện lần đầu tiên được đề xuất bởi Cresceptron để kích hoạt mạng để chịu đựng được sự biến dạng từ nhỏ đến lớn theo một cách phân cấp, trong khi sử dụng tích chập. Thăm dò mã đã hoạt động tốt, nhưng không đảm bảo, dịch chuyển bất định ở mức điểm ảnh.

Sven Behnke vào năm 2003 dựa chỉ vào các dấu hiệu của gradient (Rprop) khi đào tạo Kim tự tháp Trừu tượng Nơ ron của mình để giải bài toán giống như tái tạo hình ảnh và định vị khuôn mặt. Các phương pháp khác cũng sử dụng đào tạo trước không có giám sát để tạo ra một mạng nơ ron, khiến nó lần đầu tiên học được bộ dò đặc điểm nói chung là hữu ích. Sau đó mạng này được đào tạo tiếp tục bằng cách truyền ngược có giám sát để phân loại dữ liệu có dán nhãn. Mô hình sâu này của Hinton và các cộng sự (2006) liên quan đến việc học phân phối của một đại diện cao cấp bằng cách sử dụng các lớp kế tiếp của các biến tiềm ẩn nhị phân hoặc giá trị thực. nó sử dụng một máy Boltzmann hạn chế (Smolensky, 1986) để mô hình hóa mỗi lớp mới của các đặc điểm cao cấp hơn. Mỗi lớp mới đảm bảo một sự tăng trưởng trong biên thấp của kiểm tra tỷ lệ giống của dữ liệu, do đó tăng cường cho mô hình, nếu được huấn luyện đúng cách. Một khi đã đủ nhiều lớp đã được học, kiến trúc sâu có thể được sử dụng như là một mô hình thể sinh bằng cách tái tạo dữ liệu khi lấy mẫu xuống mô hình đó (một "sự vượt qua tổ tiên") từ các kích hoạt tính năng cấp đỉnh. Hinton báo cáo rằng các mô hình của mình là trích xuất các đặc điểm hiệu quả tính theo chiều cao, cấu trúc dữ liệu. Nhóm Google Brain do Andrew Ng và Jeff Dean đã tạo ra một mạng nơ ron học cách để nhận dạng được những khái niệm cao cấp hơn, chẳng hạn như con mèo, chỉ từ xem những hình ảnh không được dán nhãn từ các video trên YouTube.

2.3.4. Một số mô hình nhận dạng đối tượng sử dụng kỹ thuật học máy, học sâu

a) Mô hình thuật toán CNN, RCNN, Faster RCNN

CNN (Convolutional Neural Network – mạng nơ ron tích chập) 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. CNN có các bản cải tiến như R-CNN (Region with CNN feature) và Faster R-CNN.

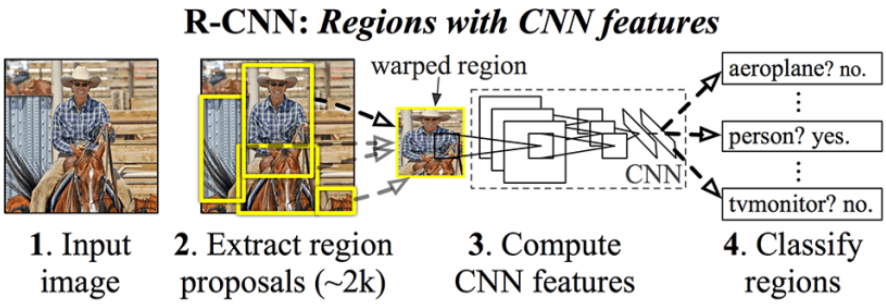
RCNN

- R-CNN sử dụng một thuật toán gọi là Selective search (Tìm kiếm chọn lọc) để đưa ra các Bounding boxes (Hộp giới hạn), hay còn gọi là Region proposals, chứa các vùng có thể có vật thể ở trong.
- R-CNN sử dụng các mạng đã được huấn luyện sẵn như Alex-net, VGG-16 để tính toán feed-forward, sau đó huấn luyện SVM (Super Vector Machine) để xác định được vật thể nào được chứa trong vùng đề nghị đó.
- R-CNN sử dụng thuật toán hồi quy tuyến tính (Linear Regression) để hiệu chỉnh các giá trị (vị trí các đỉnh) của vùng đề nghị.

Faster RCNN

- Sử dụng các mạng huấn luyện sẵn để feed-forward các vùng đề nghị, sẽ tốn nhiều thời gian bởi với mỗi ảnh thuật toán tìm kiếm chọn lọc sẽ cho ra hàng nghìn vùng đề nghị.
- Thuật toán sẽ chỉ feed-forward một lần đối với ảnh gốc, thu được đặc trưng tích chập của ảnh đó. Ví dụ với một hình ảnh có kích thước $600*600*3$, ta sẽ thu được đặc trưng với kích thước $37*37*512$. Kích thước của đặc trưng bị giảm nhỏ khoảng 16 lần

- Dựa vào kích thước cùng vị trí của các vùng đề nghị đối với ảnh gốc, ta sẽ tính toán được vị trí của vùng đề nghị trong đặc trưng tích chập.
- Sử dụng giá trị đặc trưng của vùng đề nghị ta dự đoán được vị trí các đỉnh của hộp giơi hạn cũng như vật thể nằm trong hộp giới hạn là gì.



Hình 2.12: Mô hình thuật toán R-CNN

Đối với Fast RCNN, do chia sẻ tính toán giữa các vùng trong ảnh, tốc độ thực 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 vùng đề nghị đầu vào, chỉ có thể thực thi tuần tự trên CPU. Faster RCNN giải quyết vấn đề này bằng cách sử dụng DNN (Deep neural network – Mạng neutron sâu) để tính toán các vùng đề nghị này.

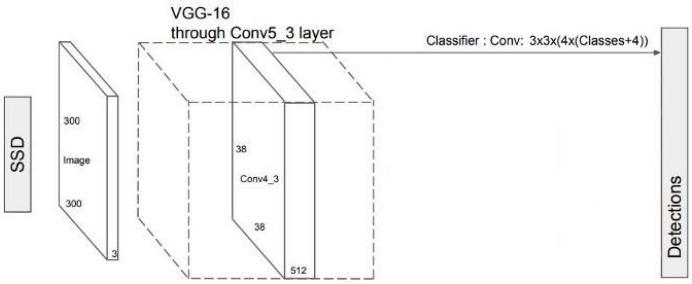
b) Mô hình thuật toán SSD (Single shot object detectors)

SSD (Single shot object detectors – phát hiện đối tượng đơn) được thiết kế để phát hiện đối tượng trong thời gian thực. R-CNN nhanh hơn sử dụng mạng đề xuất khu vực để tạo các hộp ranh giới và sử dụng các hộp đó để phân loại các đối tượng. Mặc dù được coi là khởi đầu chính xác, toàn bộ quá trình chạy ở 7 khung hình mỗi giây. Thấp hơn nhiều so với những gì cần cho một nhu cầu xử lý thời gian thực. SSD tăng tốc quá trình bằng cách loại bỏ sự cần thiết của mạng đề xuất khu vực. Để giải quyết về vấn đề độ chính xác giảm, SSD áp dụng một vài cải tiến bao gồm các feature map đa kích thước và sử dụng các hộp mặc định. Những cải tiến này cho phép SSD tiến gần được với độ chính xác của Faster R-CNN nhưng lại có thể sử dụng hình ảnh có độ phân giải thấp hơn, giúp đẩy tốc độ cao hơn.

Mô hình SSD được chia làm 2 giai đoạn:

- Trích xuất bản đồ đặc trưng (dựa vào mạng cơ sở VGG16) để tăng hiệu quả trong việc phát hiện => thì nên sử dụng ResNet, InceptionNet, hoặc MobileNet
- Áp dụng các bộ lọc tích chập để có thể detect được các đối tượng.

Giai đoạn 1: Mô hình SSD sử dụng mạng cơ sở VGG16 để trích xuất feature map



Hình 2.13: Giai đoạn 1 - Mô hình thuật toán SSD

- Đầu vào: image kích thước (300 * 300)
- Sau đó sử dụng VGG16 làm mạng cơ sở để trích xuất bản đồ tính năng. Lưu ý, đối với VGG16, chúng ta sẽ sử dụng lớp Conv4_3 là một trong những lớp predictor object của mạng, những lớp còn lại chỉ có tác dụng trích xuất các đặc trưng của hình ảnh đầu vào
- Hình ảnh trên là sự minh họa cho Conv4_3 với kích thước kernel là (8, 8) cho nên ko gian kích thước là (38, 38).

Sau khi tạo ra feature map thì mỗi ô của bản đồ đặc trưng sẽ đưa ra 4 dự đoán đối tượng. Mỗi dự đoán này bao gồm:

- Bounding box (Hộp xác định ranh giới)
- 21 score biểu thị cho độ tin cậy của mỗi lớp (bao gồm cả lớp nền)

– score: là xác suất xuất hiện đối tượng được dự đoán có trong bounding box, 20 score là 20 xác suất phát hiện 20 loại đối tượng trong thực tế, 1 score còn lại là xác suất dự đoán trong hộp ranh giới là lớp nền

Do đó, lớp Conv4_3 đưa ra tổng số lượng dự đoán là 38 × 38 × 4 (4 bbx được dự đoán tại mỗi vị trí pixel). Mỗi dự đoán bao gồm một bbx và 21 score biểu thị cho 21 class ~ 21 score này là 21 xác suất thể hiện cho khả năng trong bbx là đối tượng thuộc class đó.

Giai đoạn 2: Sử dụng trình dự đoán tích chập (Convolution preditors) cho phát hiện đối tượng

Sau khi trích xuất bản đồ đặc trưng dựa vào kiến trúc của mạng cơ sở (Base network), mô hình SSD sẽ tiếp tục tính toán cả hai giá trị: đối với từng vị trí và điểm cho các lớp bằng việc sử dụng các bộ lọc tích chập nhỏ. Sau khi giải nén bản đồ đặc trưng, SSD sử dụng một kernel 3 × 3 trên mỗi ô để tạo ra dự đoán. Mỗi đầu ra của bộ lọc là 25 kênh: 21 điểm đối với mỗi lớp + 1 bbx.

Áp dụng bộ lọc 3 × 3 để tạo ra các dự đoán đối với mỗi vị trí và các lớp. Quá trình sử dụng các trình phát hiện tích chập đơn giản cũng chỉ là quá trình predict ra các bbx dự đoán (với mỗi vị trí pixel sẽ là 4 bbx dự đoán giống như VGG16 đang làm) có điều nó khác bởi vì nó sử dụng các feature map có kích thước khác nhau với mục đích như sau:

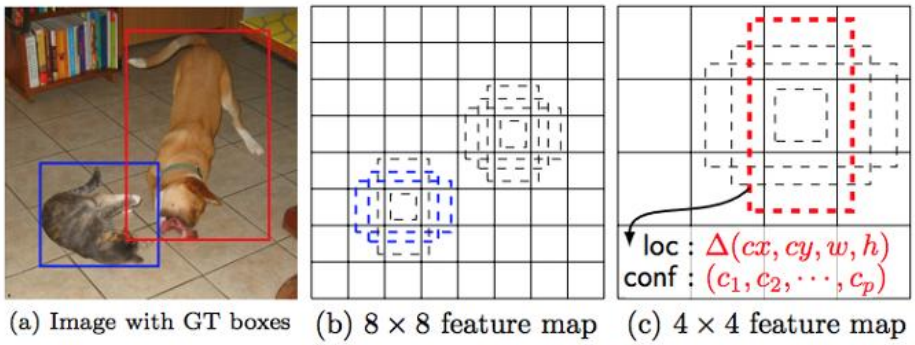
- Bản đồ đặc trưng nhiều tỷ lệ cho việc nhận dạng.

  Vấn đề phối cảnh sẽ thay đổi kích thước của các vật thể

  - Ở trên là việc phát hiện đối tượng ở 1 kích thước, trên thực tế thì việc detect nhiều đối tượng vs nhiều kích thước sẽ sử dụng quy trình tương tự nhưng ở các lớp khác nhau
  - Đối với các bản đồ đặc trưng có độ phân giải thấp ở bên phải sẽ có khả năng phát hiện được đối tượng có kích thước khác nhau.

- Bbx mặc định: có thể hiểu là các bbx xếp chồng lên nhau. Nó giúp phát hiện các đối tượng
- Bbx bao gồm nhiều loại đối tượng hơn. Trong thực tế, thì ko phải hình dạng của các đối tượng là tùy ý, nó luôn tuân theo một quy tắc, tỷ lệ nào đó, cho nên ta có thể sd một bbx default mặc định như sau, nó có khả năng giúp bao và phát hiện được nhiều đối tượng nhất: Đối với mỗi lớp bản đồ đặc trưng, nó sẽ chia sẻ cùng một bộ bbx mặc định khác nhau để có thể tùy chỉnh nhận dạng các đối tượng có các kích thước khác nhau (các bản đồ đặc trưng ở các độ phân giải khác nhau). 4 hộp bbx màu xanh dưới đây là 4 hộp bbx mặc định của một lớp

Dưới đây là ví dụ cho việc sử dụng bản đồ đặc trưng nhiều tỷ lệ và bbx mặc định để có thể nhận dạng ra các đối tượng nhiều kích thước



Hình 2.14: Ví dụ cho việc sử dụng bản đồ đặc trưng nhiều tỷ lệ và bbx mặc định

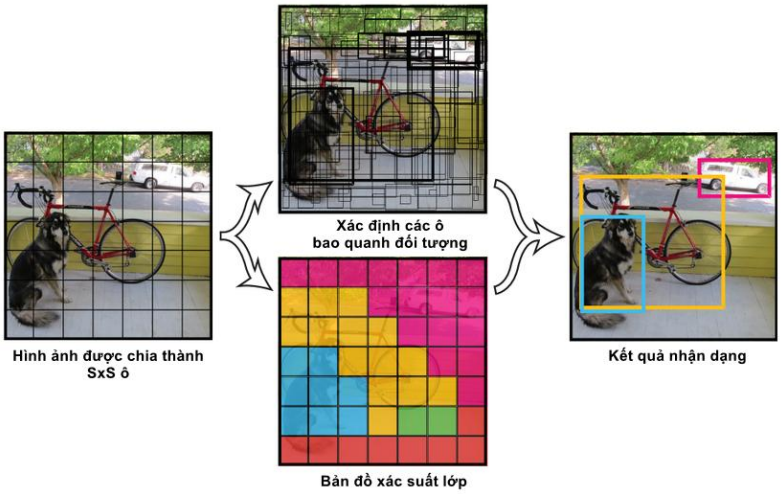
Con chó khớp với bbx mặc định màu đỏ được thể hiện trong bản đồ đặc trưng có kích thước nhỏ là 4 * 4, và bbx mặc định màu xanh phát hiện con mèo ở bản đồ đặc trưng kích thước lớn 8*8. Do đó, bản đồ đặc trưng ở độ phân giải cao thì phát hiện được đối tượng nhỏ và bản đồ đặc trưng ở độ phân giải thấp thì phát hiện được đối tượng lớn.

Nhược điểm: Khi hình ảnh đầu vào cho vào SSD thì kích thước của hình ảnh đã giảm đi đáng kể cho nên các độ chính xác nhận dạng đối tượng không cao.

c) Mô hình thuật toán YOLO (You only look once)

Mô hình YOLO (You only look once) là một mô hình CNN để nhận dạng đối tượng mà một ưu điểm nổi trội là nhanh hơn nhiều so với những mô hình cũ. Thậm chí có thể chạy tốt trên những thiết bị IoT như raspberry pi.

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



Hình 2.15: Mô tả hoạt động của mô hình thuật toán YOLO

Với đầu vào là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước S x S x (5 x N + M) với số lượng tham số mỗi ô là (5 × N + M) với N và M lần lượt là số lượng ô (Box) và lớp (Class) mà mỗi ô cần dự đoán.

Ví dụ với hình ảnh trên chia thành 7×7 ô, mỗi ô cần dự đoán 2 ô ranh giới và 3 đối tượng: con chó, ô tô, xe đạp thì đầu ra là 7×7×13, mỗi ô sẽ có 13 tham số, kết quả trả về 7×7×2=98) ô ranh giới.

Mỗi ô vuông bao gồm một tập các thông tin mà mô hình phải dự đoán:

- Đối tượng duy nhất mà ô vuông đó chứa. Tâm của đối tượng cần xác định nằm trong ô vuông nào thì ô vuông đó chứa đối tượng đó. Chúng ta có thể tăng kích

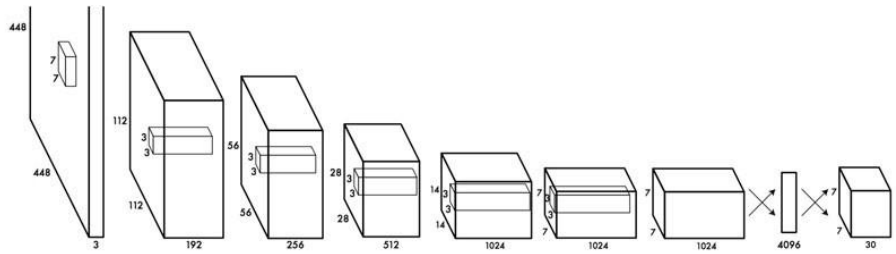
thước lưới từ 7x7 lên kích thước lớn hơn để có thể xác định được nhiều đối tượng hơn. Ngoài ra, kích thước của ảnh đầu vào phải là bội số của kích thước lưới.

- Mỗi ô vuông chịu trách nhiệm dự đoán 2 bounding box của đối tượng. Mỗi ô ranh giới cần dự đoán có chứa đối tượng hay không và thông tin vị trí của ô ranh giới gồm trung tâm ô của đối tượng và chiều dài, rộng của ô đó. Một điều cần lưu ý, lúc cài đặt chúng ta không dự đoán giá trị pixel mà cần phải chuẩn hóa kích thước ảnh về đoạn từ [0-1] và dự đoán độ lệch của tâm đối tượng đến ô chứa đối tượng đó

Tổng hợp lại, với mỗi ô vuông chúng ta cần dự đoán các thông tin sau:

- Ô vuông đó có chứa đối tượng nào hay không
- Dự đoán độ lệch 2 ô chứa đối tượng so với ô vuông hiện tại
- Lớp của đối tượng đó

Như vậy với mỗi ô vuông chúng ta cần dữ đoán một vector có (nbox+4*nbox+nclass) chiều.



Hình 2.16: Ví dụ xây dựng mô hình CNN cho YOLO với kích thước lưới là 7x7

Chúng ta đã cần biết phải dự đoán những thông tin nào đối với mỗi ô vuông, điều quan trọng tiếp theo là xây dựng một mô hình CNN có cho ra kết quả với kích thước phù hợp theo yêu cầu của chúng ta, tức là gridsize x gridsize x (nbox+4*nbox+nclass). Ví dụ với kích thước lưới là 7x7 là mỗi ô vuông dự đoán 2

ô, và có 3 loại đối tượng tất cả thì chúng ta phải cần đầu ra có khối 7x7x13 từ mô hình CNN.

YOLO sử dụng Linear regression (Hồi quy tuyến tính) để dự đoán các thông tin ở mỗi ô vuông. Do đó, ở lớp cuối cùng chúng ta sẽ không sử dụng bất kì hàm kích hoạt nào cả. Với ảnh đầu vào là 448x448, mô hình CNN có 6 tầng với kích thước 2x2 sẽ giảm 64 lần kích thước ảnh xuống còn 7x7 ở kết quả đầu ra. Đồng thời thay vì sử dụng tầng toàn bộ kết nối ở các tầng cuối cùng, chúng ta có thể thay thế bằng tầng 1x1 với 13 bản đồ đặc trưng để kích thước đầu ra dễ dàng cho ra 7x7x13.

CHƯƠNG 3. XÂY DỰNG THUẬT TOÁN NHẬN DẠNG BIỂN SOÁT PHƯƠNG TIỆN GIAO THÔNG

Trong chương này sẽ trình bày về các bước xây dựng thuật toán nhận diện biển kiểm soát phương tiện giao thông; các công sử dụng để xây dựng thuật toán;

3.1. Các công cụ sử dụng

3.1.1. Bộ thư viện mã nguồn mở OpenCV (Open Computer Vision)

Hiện nay chúng ta có thể dễ dàng tìm kiếm các công cụ phần mềm sử dụng cho các mô hình hệ thống nhận dạng tương tự hệ thống này như OpenCV, OpenALPR, OpenVino, ... Trong bài toán này tác giả đã chọn lựa sử dụng công cụ OpenCV do tính tương thích và tùy biến cao với hệ thống này.

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.

Hình 3.1: Giới thiệu về OpenCV (Open Computer Vision)

OpenCV đươc 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 hỗ trợ các ngôn ngữ như 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 người tải 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, ...

3.1.2. Ngôn ngữ lập trình sử dụng

Về ngôn ngữ lập trình sử dụng, tác giả lựa chọn ngôn ngữ lập trình Python do tính tương tác cao với hệ thống và tốc độ xử lý nhanh. Python là một ngôn ngữ lập trình thông dịch do Guido van Rossum tạo ra năm 1990. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

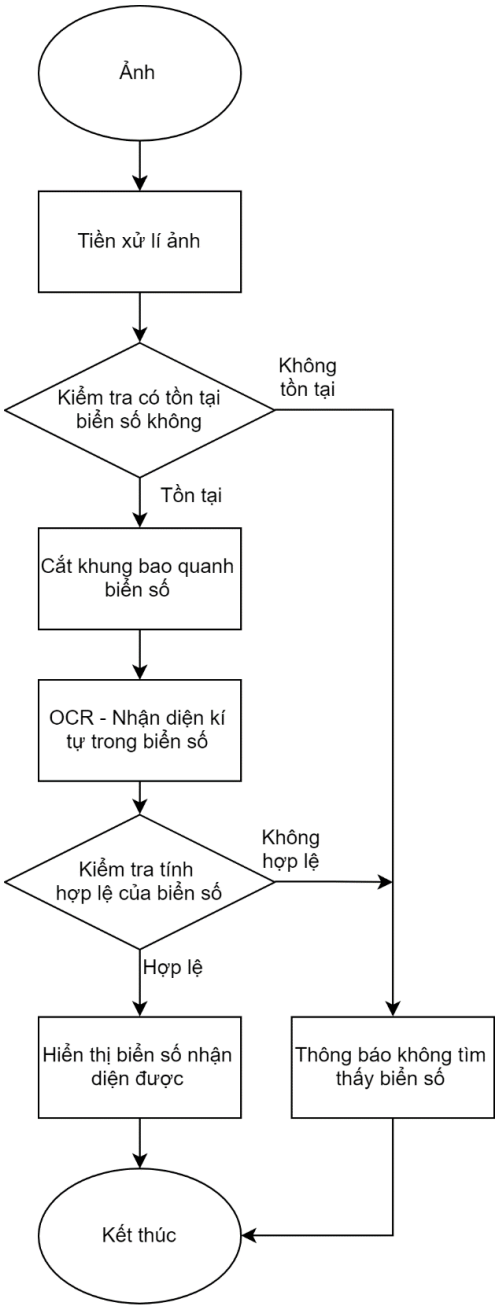
Theo đánh giá của cộng đồng, Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, như nhận định của chính Guido van Rossum trong một bài phỏng vấn ông. Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã phát triển sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Python là một ngôn ngữ được sử dụng một cách rộng rãi ở cấp cao, đa năng và là một ngôn ngữ lập trình khá năng động. Nó là ngôn ngữ "duy nhất" của khoảnh khắc và được chọn làm ngôn ngữ khởi xướng cho tất cả các khoá học bậc đại học ở khắp nơi trên thế giới. Nó có tốc độ tăng trưởng lớn nhất tính theo năm so với bất cứ ngôn ngữ lập trình nào.

Hình 3.2: Ngôn ngữ Python

Python trên toàn thế giới có tốc độ tăng trưởng lớn nhất trong 5 năm qua và năm nay, 2017, Python đã trở thành ngôn ngữ phổ biến nhất trên thế giới hiện nay, theo như PYPL (PopularitY of Programming Language) diễn đàn công bố.

3.2. Xây dựng chương trình nhận dạng biển kiểm soát từ hình ảnh

3.2.1. Lưu đồ thuật toán



Hình 3.3: Lưu đồ thuật toán nhận diện biển kiểm soát từ hình ảnh

Quá trình nhận dạng biển kiểm soát từ hình ảnh trải qua 4 giai đoạn: Tiền xử lý ảnh, tìm kiếm vị trí khả dụng của biển số trong ảnh, tìm kiếm vị trí các kí tự trong biển số và chuyển hình ảnh ký tự thành văn bản.

3.2.2. Giai đoạn 1 – Tiền xử lý ảnh

Dữ liệu hình ảnh đầu vào là hình ảnh chưa được chỉnh sửa bởi bất kì phương diện nào. Đầu vào hình ảnh của hệ thống này là các tệp ảnh thu nhận từ máy quét, máy ảnh, thiết bị ghi hình hay hay các thiết bị thu hình khác. Các hình ảnh này thường có chất lượng thấp (Bị lẫn các nhiễu, mất các chi tiết của đối tượng hay bị lệch một góc bất kỳ). Nguyên nhân là do thiết bị thu không đảm bảo, điều kiện thu không đảm bảo hay mất mát dữ liệu trong quá trình truyền tải, sao chép thông tin. Để các bước tiếp theo thu được kết quả tốt, cần phải có quá trình tiền xử lý. Quá trình này bao gồm các công đoạn khôi phục và tăng cường chất lượng hình ảnh:

- Khôi phục hình ảnh nhằm mục đích loại bỏ hay giảm thiểu các ảnh hưởng của môi trường tác động lên ảnh. Bao gồm các bước lọc ảnh, khử nhiễu, xoay ảnh, nhằm giảm thiểu các biến dạng của ảnh và đưa ảnh về trạng thái gần như ban đầu.
- Tăng cường ảnh không phải là làm tăng lượng thông tin trong ảnh mà là làm nổi bật các đặc trưng của ảnh giúp cho việc xử lý phía sau được hiệu quả hơn. Công đoạn này bao gồm các việc như: lọc độ tương phản, làm trơn ảnh, nhị phân ảnh.

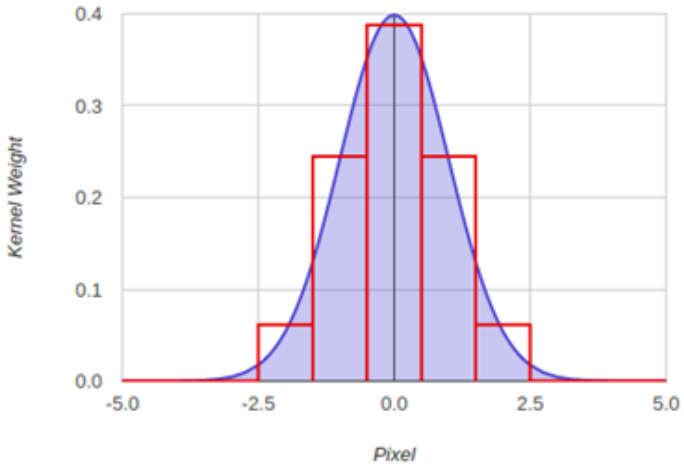
Trong thực tế có rất nhiều phương pháp tiền xử lý ảnh khác nhau, nhưng trong khuôn khổ bài toán này. Ta sẽ sử dụng các phương pháp chính đó là: Làm mờ ảnh, biến đổi ảnh xám và nhị phân ảnh.

Tiền xử lý ảnh - Làm mờ ảnh

Phép làm mờ ảnh được sử dụng rất nhiều trong xử lý ảnh và có vai trò rất quan trọng. Hiệu ứng làm mở mang lại kết quả như:

- Giảm nhiễu trong ảnh
- Làm trơn ảnh. Việc làm trơn ảnh sẽ giảm sắc nét của cạnh, thay vào đó vùng trơn sẽ được lan ra

Trong thực tế có rất nhiều phương pháp làm mờ ảnh khác nhau, tùy vào thuật toán và các bộ lọc. Trong bài toán này ta sẽ sử dụng bộ lọc mờ Gaussian (Gaussian Filter). Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập 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 3.4: Biểu diễn ma trận lọc Gaussian

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.

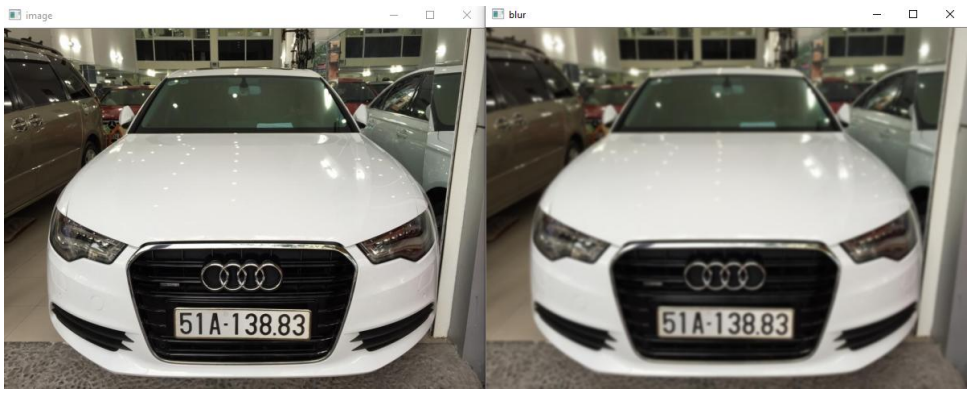
Trên thực tế, việc lọc ảnh dựa trên hàm Gauss 2 chiều (ngang và dọc). Phân phối chuẩn 2 chiều có thể biểu diễn dưới dạng:

$$G\_0(x,y) = Ae^{\frac{-(x-\mu x)^2}{2\sigma^2 x}+\frac{-(x-\mu y)^2}{2\sigma^2 y}}$$

Trong đó:

- μ là trung bình (đỉnh)
- σ^2 là phương sai của các biến số x và y.

Tham số μ quyết định tác dụng của bộ lọc Gauss lên ảnh. Độ lớn của ma trận lọc cần được lựa chọn cho đủ rộng.



Hình 3.5: Làm mờ ảnh bằng bộ lọc Gaussian

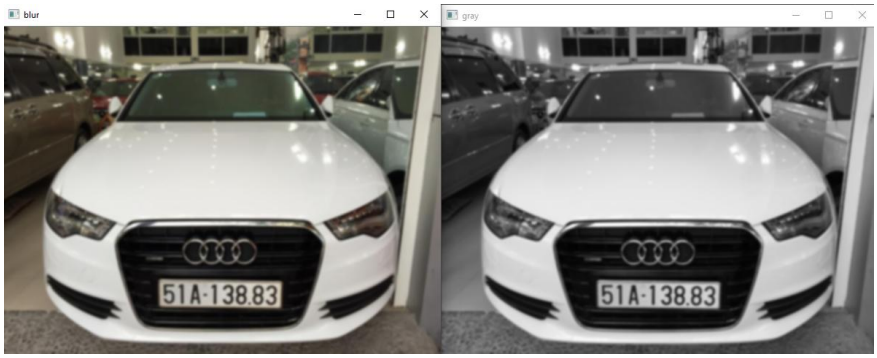
Tiền xử lý ảnh - Biến đổi ảnh xám

Trong xử lý ảnh việc chuyển đổi ảnh màu sang ảnh xám là công việc vô cùng phổ biến. Ảnh màu thực chất chỉ là tập hợp của những ma trận số có cùng kích thước. Khi muốn xử lý thông tin trên ảnh, sẽ dễ dàng hơn nếu ta chỉ xử lý dữ liệu trên một ma trận số thay vì nhiều ma trận số. Việc biến đổi ảnh màu về ảnh số (Grayscale converting) xuất hiện vì mục đích trên - biến đổi thông tin ảnh về một ma trận số hai chiều duy nhất.

Giả sử, hình ảnh được lưu trữ dưới dạng RGB. Điều này có nghĩa ta có ba ma trận xám tương ứng cho màu R, G, B. Công việc cần làm là tìm cách tổng hợp ba ma trận này về thành một ma trận duy nhất. Một trong số các công thức phổ biến để thực hiện việc đó là: Y = 0.2126R + 0.7152G + 0.0722B

Trong đó:

- Y: Ma trận xám cần tìm
- R: Ma trận xám đỏ của ảnh
- G: Ma trận xám lục của ảnh
- B: Ma trận xám lam của ảnh



Hình 3.6: Biến đổi ảnh xám

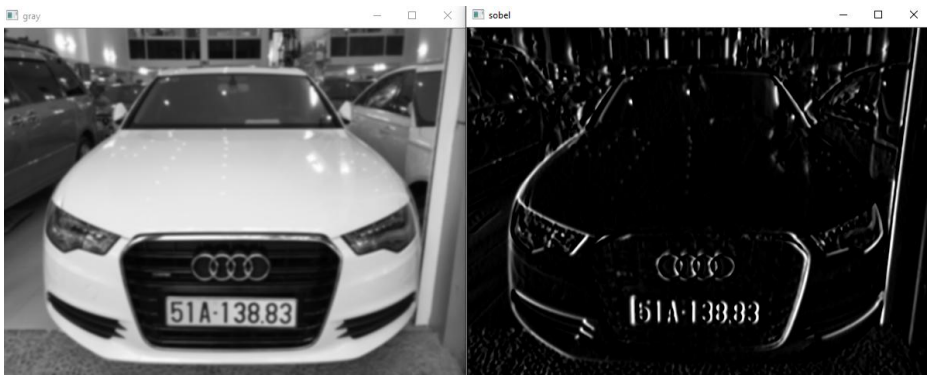
Tiền xử lý ảnh - Làm nổi biên ảnh, sử dụng kỹ thuật Sobel

Phép Sobel là sự kết hợp giữa làm mịn Gauss và phép vi phân, do vậy nó ít bị ảnh hưởng bởi nhiễu. Việc kết hợp này không hẳn là việc lọc nhiễu bằng phép Gauss trước, rồi thực hiện Sobel để tìm biên mà phép Gauss và Sobel sẽ được kết hợp để tạo ra một ma trân lọc rồi sau đó nhân chập ma trận này với ảnh.

Xét một hàm mức xám f, ma trận lọc Gauss h, ta có công thức

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x}h\right) * f \qquad \text{(x.x)}$$

Như vậy, thay vì áp dụng bộ lọc Gauss lên ảnh (kích thước khá lớn) rồi áp dụng lọc Sobel để tìm biên, ta có thể áp dụng phép Sobel lên ma trận Gauss (kích thước nhỏ) rồi sau đó nhân chập ma trận thu được với ảnh để cho ra kết quả tương tự. Việc này sẽ giảm đáng kể chi phí tính toán.



Hình 3.7: Làm nổi biên ảnh sử dụng kĩ thuật Sobel

Nhị phân hóa ảnh

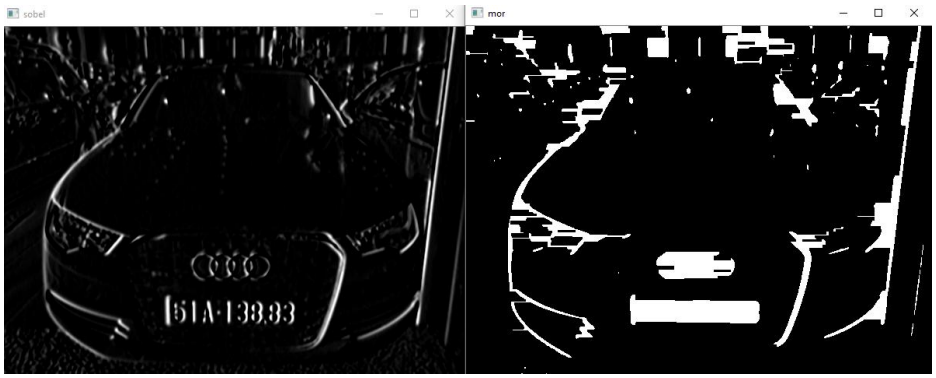
Ảnh nhị phân (hay còn gọi là binary image trong tiếng Anh) là ảnh đen trắng chỉ có 2 giá trị là 0 và 255 (miền số nguyên) hoặc 0 và 1 (miền số thực / đối với ROI).

Nhị phân hóa là quá trình biến đổi một ảnh xám thành ảnh nhị phân:

- Gọi giá trị cường độ sáng tại một điểm ảnh là I(x,y)
- INP(x,y) là cường độ sáng của điểm ảnh trên ảnh nhị phân .
- (Với 0 < x < chiều rộng) và (0 < y < chiều cao).

Để biến đổi ảnh xám thành nhị phân ta so sánh giá trị cường độ sáng của điểm ảnh với một ngưỡng nhị phân T.

- Nếu I(x,y) > T thì INP(x, y) = 0
- Nếu I(x,y) > T thì INP(x, y) = 255



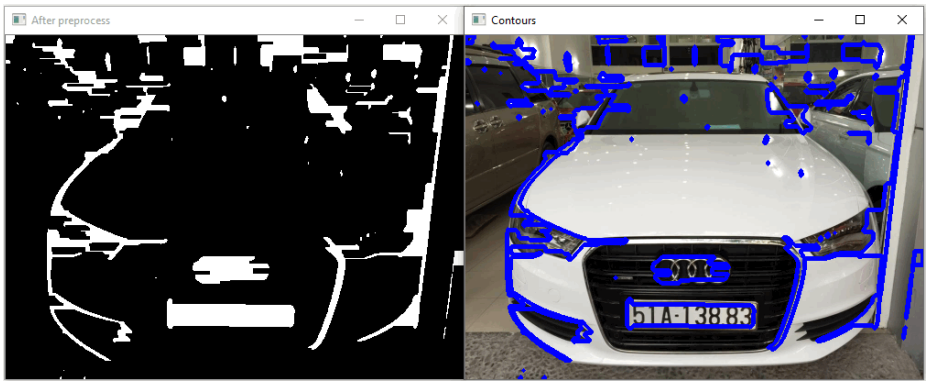
Hình 3.8: Nhị phân hóa ảnh

3.2.3. Giai đoạn 2 – Tìm vị trí khả dụng của biển số

Để xác định được các vị trí khả dụng của biển số có trong hình ảnh, bước đầu tiên ta cần tìm các vùng biên có trong ảnh.

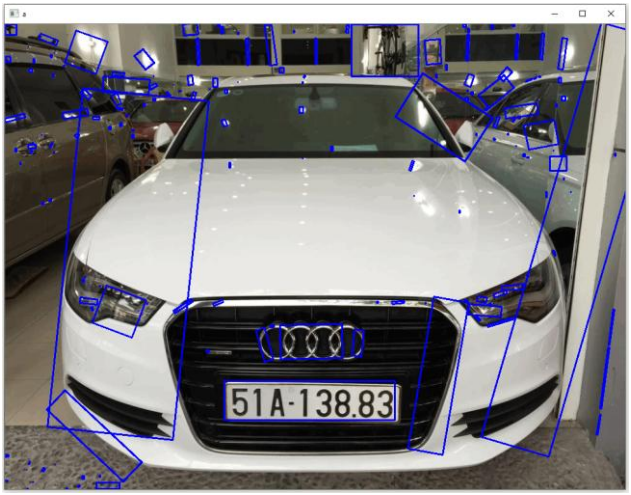
Thuật ngữ vùng biên có thể hiểu là tập các điểm liên tục tạo thành một đường cong và không có khoảng hở trong đường cong đó. Đặc điểm chung trong một vùng biên là các các điểm có cùng gần xấu xỉ một giá trị màu, hoặc cùng mật độ. Vùng

biên là một công cụ hữu ích được dùng để phân tích hình dạng đối tượng, phát hiện đối tượng và nhận dạng đối tượng.



Hình 3.9: Tìm các đường biên khả dụng

Sau khi tìm được các đường biên khả dụng, ta sẽ liên kết các đường biên này lại tạo thành những khối chữ nhật



Hình 3.10: Tạo các khối chữ nhật từ đường biên khả dụng

Với mỗi khối hiện tại sẽ bao bọc được một thành phần trong ảnh, trong đó có thể có biển số. Bước tiếp theo ta sẽ kiểm tra tính hợp lệ của các khối để dự đoán xem khối nào đang chứa biển số.

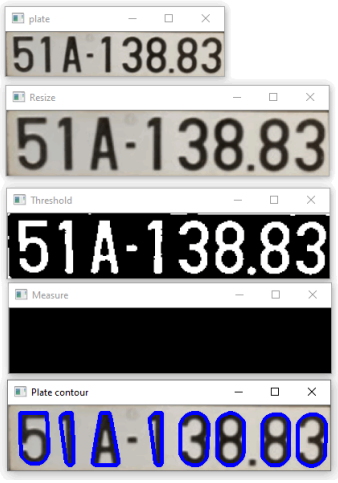
Bằng cách kiểm tra các điều kiện thích hợp cho từng khối bằng các đặc trưng trích chọn. Ví dụ như biển số phía trước xe ô tô có chiều cao là 110mm, chiều dài là 470mm gấp gần 5 lần chiều cao, như vậy ta sẽ liệt kê vào điều kiện của tập đặc trưng đó là

chiều dài phải có kích thước gấp 4 lần trở lên và không được quá 6 lần. . . . Tương tự như vậy với các đặc trưng khác, ta sẽ có một tập đặc trưng để so sánh.

Hình 3.11: Kết quả sau khi lọc bằng tập đặc trưng

3.2.4. Giai đoạn 3 – Tìm kiếm vị trí các kí tự trong biển số

Tương tự với thuật toán tìm kiếm vị trí khả dụng của biển số trong ảnh, việc tìm kiếm vị trí khả dụng của các kí tự trong một khung biển số đã xác định là một thuật toán hoàn toàn tương tự. Điểm khác biệt trong thuật toán này đó là những đặc trưng so sánh nhằm phân biệt kí tự với các thành phần khác. Trong bước tiền xử lý này ta cũng thực hiện tương tự các bước: Làm mờ, biến đổi ảnh xám, làm nổi biên, nhị phân hóa, tách biên và cuối cùng là so sánh được trưng. Kết thúc bước tiền xử lý ta sẽ thu được vị trí khung của các ký tự có trong biển số. Các bước được mô tả như hình dưới.



Hình 3.12: Tiền xử lý biển số trong nhận dạng ký tự trên biển số

Kết thúc quá trình này, ta thu được một mảng chứa các khung bao quanh ký tự. Bước tiếp theo là đưa các ký tự này vào chương trình nhận dạng kí tự.

3.2.5. Giai đoạn 4 – Chuyển đổi hình ảnh ký tự thành văn bản

Chuyển đổi hình ảnh ký tự thành văn bản hay nhận dạng ký tự quang học OCR (Optical character recognition) là một kỹ thuật được sử dụng để trích xuất văn bản từ hình ảnh hoặc quét tài liệu. Văn bản này được sử dụng để chế biến tiếp như nó có thể được chỉnh sửa, định dạng, tìm kiếm, lập chỉ mục và tự động dịch hoặc chuyển đổi sang ngôn luận.

Để xây dựng được một hệ thống nhận diện ký tự cần trải qua hai quá trình chính, quá trình thứ nhất là huấn luyện tập dữ liệu và quá trình thứ hai là dựa vào tập dữ liệu đã được huấn luyện để nhận diện ra ký tự.

Nhận diện ký tự - Huấn luyện tập dữ liệu

Có nhiều phương pháp để huấn luyện dữ liệu như học có giám sát (Supervised learning), học không giám sát (Unsupervised learning), học bán giám sát (Semi-Unsupervised learning), học củng cố (Reinforcement learning), ... Trong khuôn khổ bài toán này ta sẽ sử dụng phương pháp học có giám sát để huấn luyện tập dữ liệu.

Học có giám sát là thuật toán dự đoán đầu ra của một dữ liệu mới dựa trên các cặp đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào $X = \{x_1, x_2, ..., x_N\}$ và một tập nhãn tương ứng $Y = \{y_1, y_2, ..., y_N\}$. Trong đó x_i và y_i là các vector. Các cặp dữ liệu biết trước (xi, yi) € X x Y được gọi là tập đã huấn luyện. Từ tập dữ liệu đã được huấn luyện này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử của tập X sang một phần tử xấp xỉ của tập Y.

$$Yi \approx f(xi) \text{ với mọi } i = 1,2, ..., N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu X mới, chúng ta có thể tìm được một nhãn tương ứng y = f(x)

Ứng dụng học có giám sát trong bài toán nhận diện ký tự này, ta có hàng nghìn ví dụ của mỗi ký tự được viết bởi nhiều phông chữ khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào. Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

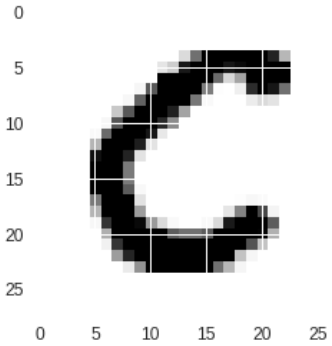
Do điều kiện thu thập bộ dữ liệu gặp nhiều hạn chế nên trong bài toán này ta sẽ sử dụng một bộ dữ liệu đã được thu thập từ trước. Bộ dữ liệu này bao gồm hình ảnh của 26 ký tự tiếng Anh và các số từ 0 đến 9.

Bộ dữ liệu này là một tập tin CSV có dung lượng 600MB bao gồm 372451 dữ liệu hình ảnh. Với mỗi dữ liệu tương ứng với một dòng. Cột đầu tiên là các số từ 0-25 đại diện cho 26 ký tự tiếng Anh. 784 cột tiếp theo là dữ liệu của một hình ảnh có kích thước 28*28. Do đó độ dài của mỗi hàng là 785.

	IY	IZ	JA	JB	JC	JD	JE	JF	JG	JH	JI	JJ	JK	JL	JM	JN	JO	JP	JQ	JR
16310	0	0	160	243	255	193	64	0	0	55	177	255	198	67	0	0	0	0	0	0
16311	0	0	0	73	255	255	90	0	0	0	0	0	0	21	128	247	223	39	0	0
16312	0	184	250	255	98	7	0	0	0	0	0	19	117	242	241	154	0	0	0	0
16313	0	0	0	0	0	0	1	133	251	155	20	0	0	0	45	190	241	110	13	0
16314	0	0	0	0	0	1	152	255	255	70	11	52	235	255	123	0	0	0	0	0
16315	0	0	0	0	235	255	255	0	0	0	20	197	255	255	255	0	0	0	0	0
16316	0	0	0	0	0	4	62	113	229	255	249	89	0	0	117	255	255	220	12	0
16317	61	232	255	255	226	65	0	0	0	0	49	227	242	89	0	0	0	0	0	0
16318	0	0	0	0	77	226	255	233	5	0	0	0	0	54	255	255	94	0	0	0
16319	0	0	0	0	0	243	255	222	51	75	132	132	190	214	247	242	121	20	0	0
16320	0	0	0	0	0	0	123	234	213	90	0	0	0	85	255	250	184	0	0	0
16321	0	0	0	0	109	221	237	100	0	51	172	216	100	0	0	0	0	0	0	0
16322	0	0	64	255	255	64	0	0	0	0	0	0	18	116	255	255	219	93	0	0
16323	0	0	0	32	62	0	0	49	245	78	0	151	225	0	0	0	0	0	0	0
16324	0	0	0	0	146	255	36	0	0	0	0	161	255	0	0	0	0	0	0	0
16325	0	0	128	231	100	8	0	29	67	41	0	0	8	86	226	236	142	0	0	0
16326	0	0	0	0	0	0	4	128	198	89	0	6	49	152	239	180	33	0	0	0
16327	0	51	246	74	0	0	0	0	0	0	0	0	0	0	0	1	87	226	143	0
16328	0	0	0	0	0	0	0	142	255	228	110	35	127	217	238	147	0	0	0	0
16329	0	0	0	0	0	206	255	146	0	0	0	0	0	0	0	126	251	255	83	0
16330	255	255	116	0	0	0	0	0	0	0	0	118	255	232	0	0	0	0	0	0

Hình 3.13: Tập dữ liệu mẫu

Mỗi hàng dữ liệu là một hình ảnh ký tự có kích thước 28*28:



Hình 3.14: Một ký tự trong tập dữ liệu mẫu

Bước tiếp theo ta sẽ sử dụng thư viện Tflearn và ngôn ngữ Python cho việc xây dựng mô hình học máy.

```
# Thêm các thư viện cần thiết
import tensorflow as tf
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers import regression
from tflearn.data_utils import to_categorical
# Khởi tạo các giá trị
BATCH_SIZE = 32
IMG_SIZE = 28
N_CLASSES = 4
LR = 0.001
N_EPOCHS = 50
```

Trong đó:

- BATCH_SIZE: kích thước một lượng dữ liệu (ảnh) truyền vào

- IMG_SIZE: kích thước mỗi chiều của hình ảnh đầu vào
- N_CLASSES: Số lượng lớp mà chúng ta cần huấn luyện
- LR: Tốc độ học
- N_EPOCHS: Số lượng epoch mà ta cần huấn luyện

Mô hình mà chúng ta sử dụng ở đây bao gồm 6 lớp Convolutional và 2 lớp Fully Connected nối tiếp nhau.

```
# Cài đặt thông số cho mô hình học
tf.reset_default_graph()
network = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1]) #1
network = conv_2d(network, 32, 3, activation='relu') #2
network = max_pool_2d(network, 2) #3
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 32, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 32, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
```

```
network = fully_connected(network, 1024, activation='relu') #4

network = dropout(network, 0.8) #5

network = fully_connected(network, N_CLASSES, activation='softmax')#6

network = regression(network)

model = tflearn.DNN(network) #7
```

Trong đó:

- #1: Kích thước dữ liệu đầu vào là [None, IMG_SIZE, IMG_SIZE, 1]

- None: đại diện cho BATCH_SIZE

- IMG_SIZE: là kích thước mỗi chiều của ảnh

- 1: là số dải màu của ảnh, do chúng ta sử dụng ảnh đen trắng nên chỉ có 1 dải màu, nếu chúng ta sử dụng ảnh màu thì số dải màu mà chúng ta sử dụng là 3, đại diện cho 3 dải màu RGB.

- #2: Cài đặt lớp tích chập Convolutional

- 32: số lượng filters

- 3: kích thước filter (3x3)

- Bước nhảy được mặc định là 1

- #3: Cài đặt lớp tổng hợp

- #4: Cài đặt lớp kết nối đầy đủ (Fully-connected layer)

- #5: Cài đặt tỉ hệ học 80%

- #6: Cài đặt lớp kết nối đầy đủ đại điện cho đầu ra

- #7: Học dựa trên mô hình đã cài đặt

Để dữ liệu đầu vào được trùng khớp với mô hình đã xây dựng, chúng ta cần phải đưa dữ liệu về định dạng phù hợp như sau:

```
train_x = train_x.reshape(-1, IMG_SIZE, IMG_SIZE, 1)

val_x = val_x.reshape(-1, IMG_SIZE, IMG_SIZE, 1)

test_x = test_x.reshape(-1, IMG_SIZE, IMG_SIZE, 1)
```

Tương tự với nhãn, đưa nhãn về dạng vector:

```
original_test_y = test_y # được sử dụng để test ở bước sau

train_y = to_categorical(train_y, N_CLASSES)

val_y = to_categorical(val_y, N_CLASSES)

test_y = to_categorical(test_y, N_CLASSES)
```

Bước tiếp theo là tiến hành huấn luyện

```
model.fit(train_x, train_y, n_epoch=N_EPOCHS, validation_set=(val_x, val_y), show_metric=True)
```

Sau khi huấn luyện, kết quả thu được như sau:

```
Training Step: 52190  | total loss: 0.23665 | time: 12.616s

| Adam | epoch: 067 | loss: 0.23665 - acc: 0.9886 -- iter: 49984/50000

Training Step: 52191  | total loss: 0.21299 | time: 13.646s

| Adam | epoch: 067 | loss: 0.21299 - acc: 0.9897 | val_loss: 0.02314 - val_acc: 0.9970 -- iter: 50000/50000
```

Kết thúc quá trình huấn luyện ta sẽ thu được một tập trích chọn đặc trưng của dữ liệu. Từ tập dữ liệu này, ta sẽ dùng để nhận diện ký tự trong các bước tiếp theo.

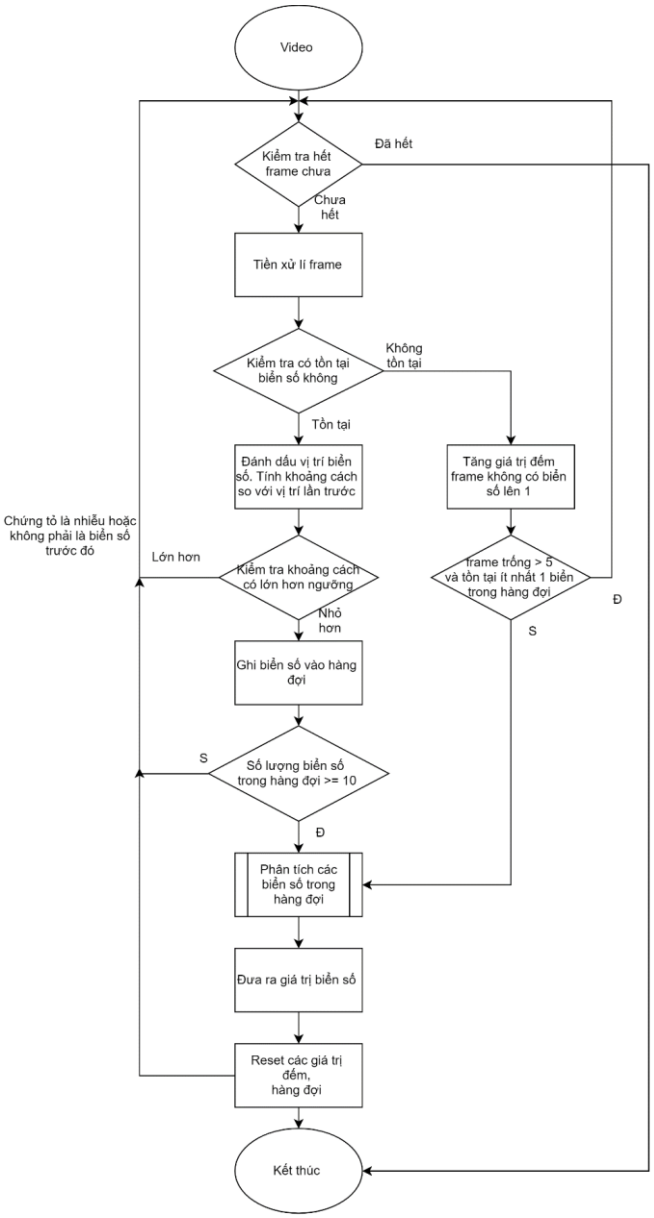
Nhận diện ký tự - Xác định ký tự dựa trên tập đặc trưng đã học

Vì các ảnh ký tự đã được tiền xử lý từ trước nên trong bước này ta không cần thực hiện tiền xử lý lại nữa mà sẽ đưa trực tiếp vào chương trình nhận diện kí tự. Ở đây ta sử dụng chính tập đặc trưng trích chọn mà ta đã huấn luyện trong bước trước.

```
input_name = "import/input"

output_name = "import/final_result"

input_operation = self.graph.get_operation_by_name(input_name);

output_operation = self.graph.get_operation_by_name(output_name);

results = self.sess.run(output_operation.outputs[0],{input_operation.outputs[0]:
tensor})

results = np.squeeze(results)

labels = self.label

top_k = results.argsort()[-1:][::-1]

return labels[top_k[0]]
```

3.3. Xây dựng chương trình nhận dạng biển kiểm soát từ video

3.3.1. Lưu đồ thuật toán



Hình 3.15: Lưu đồ thuật toán nhận dạng biển số trong video

3.3.2. Giai đoạn 1 – Xử lý nhận diện trong nhiều khung hình

Việc nhận diện biển số trong video có phần phức tạp hơn một chút, phụ thuộc vào tốc độ lia khung hình của máy quay. Với tốc độ chậm sẽ cho ta các khung hình rời rạc chất lượng kém, khó có thể nhận diện được chữ trong biển số.

Về cơ bản các bước xử lý và nhận diện biển kiểm soát trên từng khung hình chính là quá trình nhận diện biển kiểm soát trong một bức ảnh. Đối với một video ta sẽ có một loạt các khung hình liên tiếp. Bằng cách quan sát nhiều khung hình để đưa ra kết luận về biển số sẽ cho ta độ chính xác cao hơn và giải quyết được các vấn đề video chất lượng không tốt.

Với mỗi khung hình ta thực hiện giống như với nhận diện biển kiểm soát trong ảnh giống như đã trình bày ở trên. Kết quả các nhãn ký tự thu được sẽ được lưu lại phục vụ cho quá trình suy luận phía sau.

Với hai khung hình ta sẽ tính khoảng cách giữa 2 biển số. Nếu khoảng cách giữa 2 biển số lớn hơn một ngưỡng do ta quy định thì có thể kết luận đây là nhiễu hoặc là một biển số khác bị xen vào (ngưỡng này là một hệ số tương đối, dựa trên tốc độ thực tế của phương tiện trong video). Nếu 2 khung hình liên tiếp đảm bảo về khoảng cách, chứng tỏ đây là một biển số của một phương tiện, ta sẽ lưu giá trị biển số lại.

Ta cần cài đặt số khung hình mẫu chính là số khung hình cần đọc liên tiếp để xác định biển số của phương tiện. Số lượng mẫu lấy biển số cũng mang tính tương đối, phụ thuộc vào thông số của video. Video có số khung hình trên giây FPS (Frames per second) càng cao thì ta sẽ tăng mẫu lên để độ chính xác cao. Với video có FPS thấp ta sẽ chọn một hệ số phù hợp thấp hơn.

Sau khi lấy đủ số mẫu cần thiết, ta tiến hành lọc ra những khung hình có chất lượng tốt nhất để xử lý. Việc đánh giá chất lượng dựa vào 2 tiêu chí: Kích thước (kích thước càng lớn thì chất lượng càng cao); Độ mờ (Ảnh có độ mờ càng thấp thì chất lượng càng cao)

Những hình ảnh tốt nhất sẽ được đưa vào chương trình nhận diện ký tự đã nêu ở trên. Kết quả thu được sẽ là danh sách các nhãn biển số tương ứng với từng khung hình

3.3.3. Giai đoạn 2 – Suy luận giá trị biển số

Bởi các nhãn biển số lấy từ các khung hình khác nhau nên không thể dự đoán trước được toàn bộ các nhãn này là giống nhau. Như đã đề cập ở đề bài, camera thông thường sẽ cho chất lượng hình ảnh, độ phân giải thấp, tốc độ quét thấp, chính vì vậy tại một số khung hình sẽ cho hình ảnh bị mất nét dẫn đến việc nhận diện kí tự bị sai số. Bằng cách tổng hợp kết quả từ nhiều khung hình, ta có thể suy luận ra kết quả cuối cùng với độ chính xác cao hơn. Thuật toán được mô tả như dưới đây:

Giả sử ta có tập hợp 5 khung hình với các giá trị:

90A12357	90A-235-	90A12-57	90A12351

Bằng phương pháp tính trung bình ta dễ dàng suy ra:

- Vị trí đầu tiên tất cả đều là 9 => 9
- Vị trí thứ 2 tất cả đều là 0 => 0
- Vị trí thứ 3 tất cả đều là A => A
- Vị trí thứ 4 có tần suất số 1 xuất hiện lớn nhất => 1
- Vị trí thứ 5 có tần suất số 2 xuất hiện lớn nhất => 2
- Vị trí thứ 6 có tần suất số 2 xuất hiện lớn nhất => 3
- Vị trí thứ 7 có tần suất số 2 xuất hiện lớn nhất => 5
- Vị trí thứ 8 có tần suất số 2 xuất hiện lớn nhất => 7

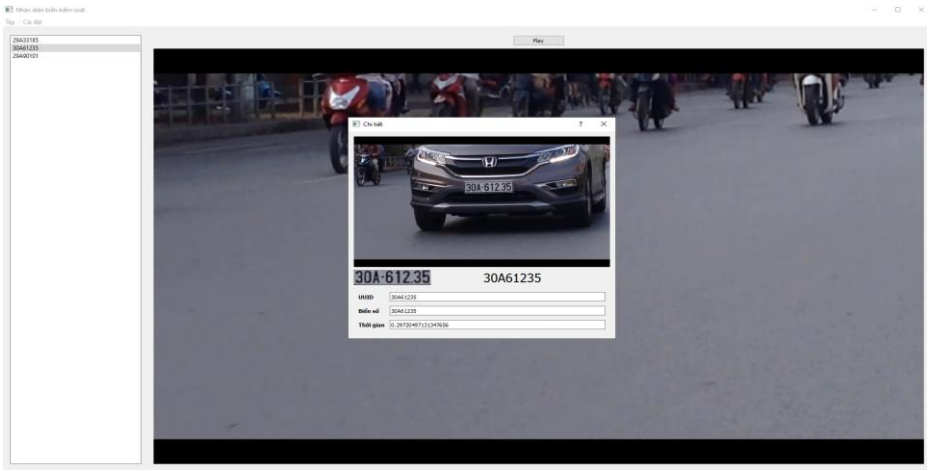
=> Kết quả cuối cùng của chương trình nhận dạng là: 90A12357

3.4. Xây dựng giao diện phần mềm mô phỏng thuật toán

Để tối ưu trong việc đưa các đầu vào là các hình ảnh và video một cách dễ dàng, cũng như xử lý các tính năng nâng cao một các thuận tiện ta sẽ xây dựng một giao diện ứng dụng ứng dụng hai thuật toán trên để xử lý. Ngoài ra ta còn có thể thông qua

giao diện phần mềm để cài đặt các thông số như tần số lấy mẫu, cài đặt lấy thời gian, nhận diện biển xe máy hoặc ô tô, ...

Giao diện được xây dựng dựa trên thư viện PyQT5 bằng ngôn ngữ Python. Ứng dụng có thể hoạt động tốt trên cả 2 nền tảng Windows và Linux.



Hình 3.16: Giao diện phần mềm nhận dạng biển kiểm soát

- Dữ liệu video thử nghiệm được thu thập trực tiếp từ hệ thống giao thông thực tế. Bộ dữ liệu thử nghiệm bao gồm 12 video được tạo thành từ 3 video gốc quay tại các thời điểm sáng sớm (ánh sáng thấp), giữa trưa (ánh sáng tốt), chiều tối (ánh sáng kém). Video gốc được quay ở độ phân giải 1920x1080, tốc độ khung hình là 60 khung hình trên giây. Bằng cách sử dụng phần mềm chỉnh sửa video, mỗi video gốc tạo ra 3 video khác có độ phân giải và tốc độ khung hình lần lượt là 1366x768 / 50fps; 800x600 / 30fps; 600x400 / 20fps. Như vậy ta có 12 video chia đều cho ba thời điểm sáng, trưa, chiều tối có thông số như sau:

Bảng 3.1: Bảng thông số video thử nghiệm kết quả phần mềm

Độ phân giải	Tốc độ khung hình
1920x1080	60fps
1366x768	50fps
800x600	30fps
600x400	20fps

Trong mỗi video đều có 10 phương tiện khác nhau di chuyển liên tục với các vận tốc khác nhau.

3.5. Nhận xét kết quả, đánh giá tính tin cậy của thuật toán

Thuật toán được chạy mô phỏng trên giao diện phần mềm xây dựng hoạt động tốt trên hai nền tảng Window và Linux. Dưới đây là kết quả đánh giá thử nghiệm thuật toán, thử nghiệm này được thực hiện toàn bộ trên một máy tính có cấu hình Intel Core I5-7200 / 2.5-2.7GHz, 8GB Ram, hệ điều hành Window.

Ưu điểm:

- Có thể nhận đầu vào bao gồm cả ảnh và video, hỗ trợ tốt tất cả các loại định dạng hình ảnh, video
- Thuật toán xử lý tiêu hao tài nguyên không đáng kể, tốc độ xử lý nhanh với tốc độ tính toán trung bình với mỗi khung hình (bao gồm xử lý biển số và nhận dạng ký tự) là 0.05s. Đánh giá dựa trên tính toán và đo đạc trực tiếp trên phần mềm thuật toán. Khi đưa một khung hình vào xử lý, ta sẽ đặt một biến giá trị lưu lại thời gian hiện tại và tại thời điểm xử lý xong khung hình cho ra kết quả, ta sẽ cập nhật lại giá trị thời gian. Và dựa vào hai mốc thời gian này ta có thể tính toán ra được thời gian cần để xử lý một khung hình. Những giá trị này được lưu lại vào một bảng giá trị. Thời gian 0.1s là giá trị trung bình dựa trên kết quả lấy mẫu 100 khung hình liên tiếp trên một video 360p với tốc độ 30 khung hình trên 1s.
- Trên một thử nghiệm khác, thuật toán có thể hoạt động tốt trên những máy tính có cấu hình thấp như Raspberry Pi 3 – một máy tính nhúng có cấu hình thấp với CPU Cortex-A53 (ARMv8) 64-bit SoC @ 1,4 GHz, Ram 1GB. Tuy nhiên đánh giá trên đây là chỉ là đánh giá riêng với thuật toán nhận diện. Còn đối với chương trình học dữ liệu lại yêu cầu một máy tính với cấu hình cao hơn. Với máy tính có cấu hình càng cao thì tốc độ học dữ liệu càng nhanh.

- Độ chính xác đạt 90% đối với các video rõ nét, giảm dần với các video có độ phân giải và chất lượng thấp hơn. Dữ liệu thử nghiệm được theo ghi lại tại bảng dưới đây:

Bảng 3.2: Bảng kết quả thử nghiệm

Thời gian	**Video**	**Kết quả**
Sáng sớm	1920x1080 / 60fps	10/10
	1366x768 / 50fps	10/10
	800x600 / 30fps	9/10
	600x400 / 20fps	8/10
Buổi trưa	1920x1080 / 60fps	10/10
	1366x768 / 50fps	10/10
	800x600 / 30fps	10/10
	600x400 / 20fps	9/10
Chiều tối	1920x1080 / 60fps	7/10
	1366x768 / 50fps	7/10
	800x600 / 30fps	5/10
	600x400 / 20fps	3/10
Tổng		98/120

Như vậy với video quay với chất lượng tốt kết quả chính xác đạt 27/30 (90%). Khi quay vào thời điểm ban ngày hay buổi sáng sớm với độ sáng trung bình và tốt sẽ cho độ chính xác tốt nhất, với kết quả ở độ phân giải thấp nhất là 17/20 đạt 85%. Còn với video quay tại buổi chiều tối với độ sáng kém cho kết quả khá thấp với 70% ở độ phân giải cao và chỉ 30% ở độ phân giải thấp.

- Có thể cài đặt các thông số như tần số lấy mẫu, nhận diện xe máy hoặc ô tô
- Phần mềm hoạt động tốt trên cả hai nền tảng window và linux.

Nhược điểm

- Chưa hoạt động tốt với các hình ảnh, video thiếu sáng hoặc quay vào buổi tối.
- Thuật toán chưa xử lý tốt khi có nhiều biển số trong một khung hình
- Phần mềm khá nặng, toàn bộ dung lượng sau khi xây dựng thành một tệp chạy duy nhất nặng 115MB

CHƯƠNG 4. KẾT QUẢ VÀ BÀN LUẬN

4.1. Kết quả

Đề tài đã hoàn thành với đầy đủ các mục đã trình bày trong đề cương được thông qua. Bao gồm các nội dung chính

- Nghiên cứu tổng quan về nhận diện biển kiểm soát phương tiện giao thông
- Nghiên cứu các vấn đề về xử lý, nhận dạng ảnh
- Xây dựng thuật toán nhận diện biểm kiếm soát phương tiện giao thông
- Chương trình thuật toán đã được tích hợp trong một giao diện phần mềm, có khả năng hoạt động trên các nền tảng Window và Linux. Sản phẩm phần mềm có khả năng áp dụng trong thực tế.

Kết quả của công trình nghiên cứu bao gồm những lý thuyết nghiên cứu được đã trình bày trong đề tài và phần mềm nhận dạng biển kiểm soát giao thông.

4.2. Bàn luận

Đề tài đã được hoàn thành song vẫn còn một vài nhược điểm chưa được giải quyết. Kết quả của đề tài sẽ là nền móng lý thuyết cho các đề tài tiếp theo với mục đích hoàn thiện công trình nghiên cứu này hơn nữa.

TÀI LIỆU THAM KHẢO

[1] Alexander Mordvintsev & Abid K. OpenCV - Python Tutorials Documentation. Nov 05-2017.

[2] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. arXiv preprint arXiv:1612.06851, 2016.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017.

[3] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox and A. Farhadi. Iqa: Visual question answering in interactive environments. arXiv preprint arXiv:1712.03316, 2017.

[4] Dmitry Batenkov. Real-Time Detection with Webcam. June 2010.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2): 303–338, 2010.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.