

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

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

HÀ NỘI - 2020

Luận văn được hoàn thành tại:
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **TS. NGUYỄN NGỌC MINH**

Phản biện 1:

.....

Phản biện 2:

.....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: giờ ngày tháng năm 2020

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

MỞ ĐẦU

Mục tiêu

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

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

- Chương 1: Tổng quan về nhận dạng biến 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 biến kiểm soát phương tiện giao thông
- Chương 4: Kết quả và bàn luận

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

1.1. Vai trò nhận dạng biểu 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 tiện giao thông.

Ở các nước trên thế giới và nước ta hiện nay, việc nhận diện biểu 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 kiểm chế tai nạn giao thông (TNGT), giảm ùn tắc giao thông. Trong đó, đề tài “Ứng dụng phần mềm nhận dạng biểu 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 ...

1.2. Thực trạng về nhận dạng biểu 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 biểu 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, Vio 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 biểu số trên thế giới.

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

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.

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 sơ 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.

Các bước cơ bản của xử lý ảnh bao gồm:

- Thu nhận ảnh
- Tiền xử lý ảnh
- Phân đoạn ảnh
- Biểu diễn ảnh sau phân đoạn
- Nhận dạng và nội suy
- Biểu diễn ảnh (hiển thị, lưu trữ)

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 rằng cơ thể mỗi loài vật chính là một hệ thống nhận dạng tối ưu nhất.

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 hỏi 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:

- Tư thế chụp, góc chụp
- Sự xuất hiện hoặc thiếu một số thành phần
- Sự biến dạng của đối tượng
- Sự che khuất
- Sự phức tạp của hình nền
- Điều kiện sáng của ảnh

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 đối tượng 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 gửi 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 thống 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

2.3.2. Giới thiệu về học máy (*Machine Learning*)

2.3.3. Giới thiệu về học sâu (*Deep Learning*)

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

- Mô hình thuật toán CNN (Convolutional Neural Network – mạng nơ ron tích chập), RCNN , Faster-CNN
- Mô hình thuật toán SSD (Single Shot Object Detector)
- Mô hình thuật toán YOLO (You only look one)

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

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.

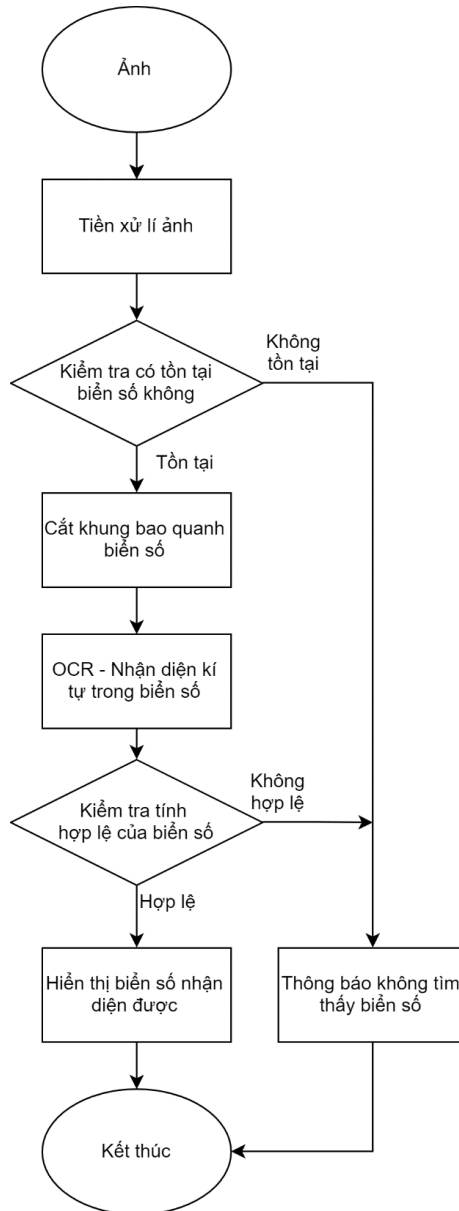
OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv hỗ trợ các ngôn ngữ như C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOS lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng người 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.

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

3.2.1. Lưu đồ thuật toán



Hình 3.1: 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

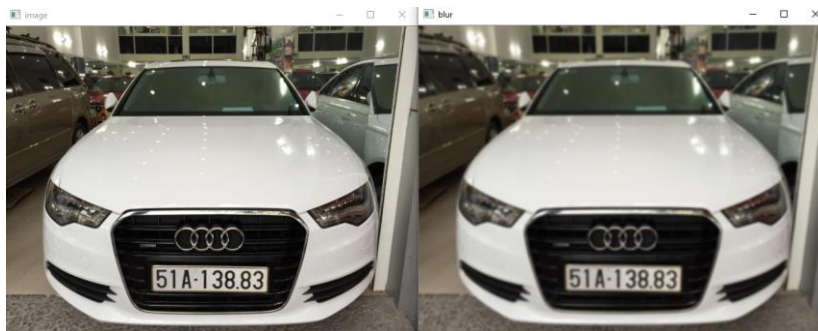
Dữ liệu hình ảnh đầu vào là hình ảnh chưa được chỉnh sửa bởi bất kì phương diện nào. Đầu vào hình ảnh của hệ thống này là các tệp ảnh thu nhận từ máy quét, máy ảnh, thiết bị ghi

hình hay 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). 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.

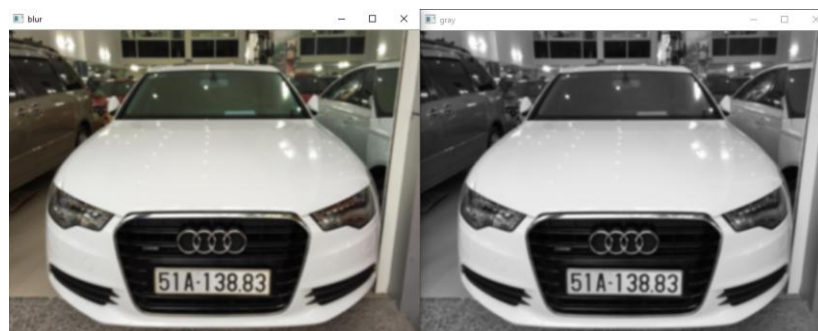
Trong thực tế có rất nhiều phương pháp tiền xử lý ảnh khác nhau, nhưng trong khuôn khổ bài toán này. Ta sẽ sử dụng các phương pháp chính đó là: Làm mờ ảnh, biến đổi ảnh xám và nhị phân ảnh.

Tiền xử lý ảnh – Làm mờ ảnh



Hình 3.2: Làm mờ ảnh bằng bộ lọc Gaussian

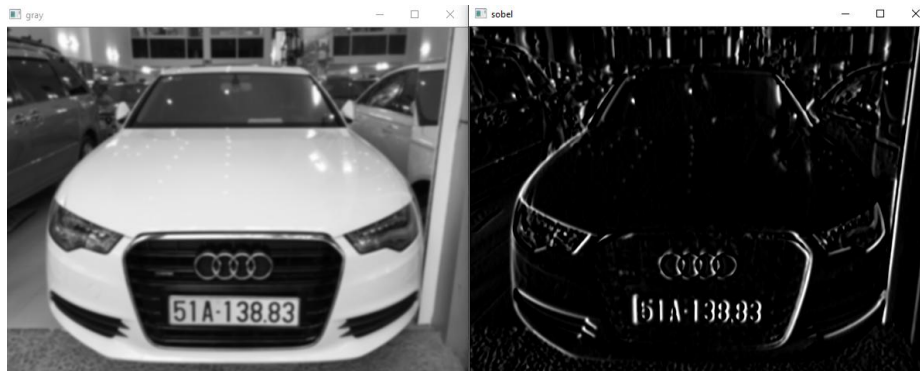
Tiền xử lý ảnh - Biến đổi ảnh xám



Hình 3.3: Biến đổi ảnh xám

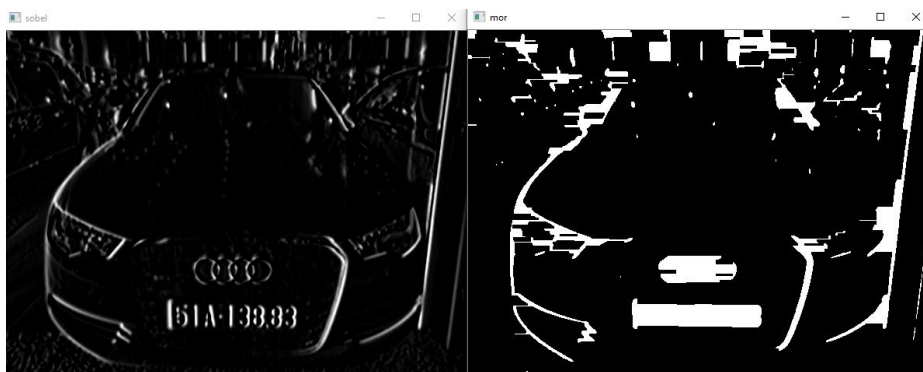
Tiền xử lý ảnh - Làm nổi biên ảnh, sử dụng kỹ thuật Sobel

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 trận lọc rồi sau đó nhân chập ma trận này với ảnh.



Hình 3.4: Làm nổi biên ảnh sử dụng kỹ thuật Sobel

Nhị phân hóa ảnh

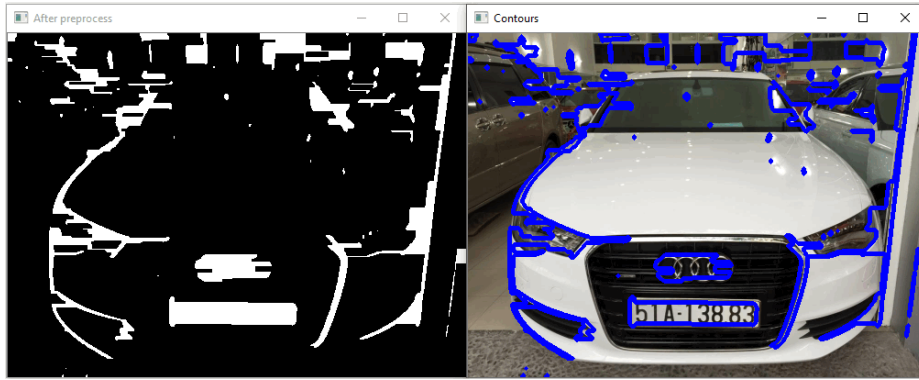


Hình 3.5: Nhị phân hóa ảnh

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

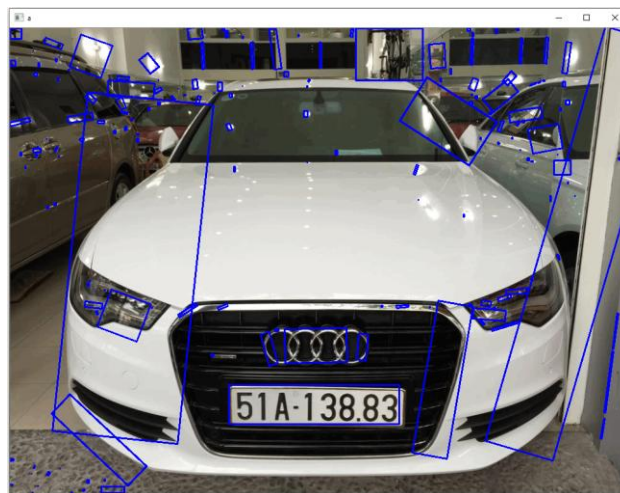
Để xác định được các vị trí khả dụng của biển số có trong hình ảnh, bước đầu tiên ta cần tìm các vùng biên có trong ảnh.

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 điểm có cùng gần xấp 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 3.6: Tìm các đường biên khả dụng

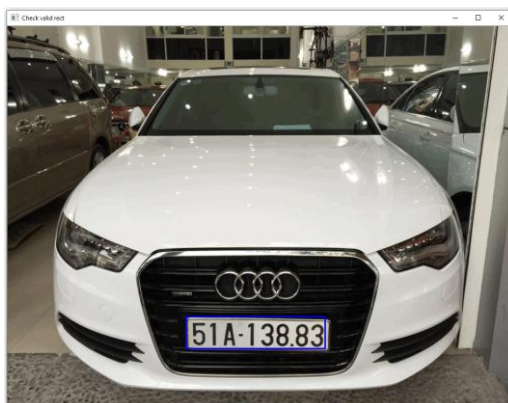
Sau khi tìm được các đường biên khả dụng, ta sẽ liên kết các đường biên này lại tạo thành những khối chữ nhật



Hình 3.7: 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 3.8: Kết quả sau khi lọc bằng tập đặc trưng

3.2.4. Giai đoạn 3 – Tìm kiếm vị trí các ký 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 ký 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 ký 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 trung. 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 3.9: 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 ký 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.

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 phong 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.

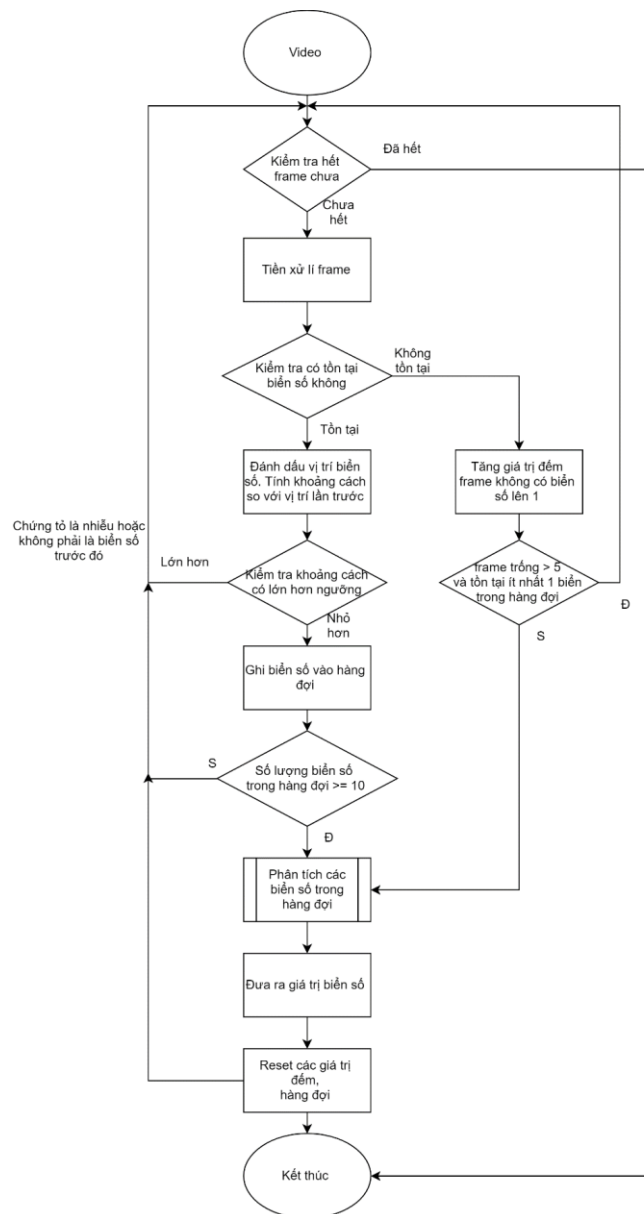
Do điều kiện thu thập bộ dữ liệu gặp nhiều hạn chế nên trong bài toán này ta sẽ sử dụng một bộ dữ liệu đã được thu thập từ trước. Bộ dữ liệu này bao gồm hình ảnh của 26 ký tự tiếng Anh và các số từ 0 đến 9. Bộ dữ liệu này là một tập tin CSV có dung lượng 600MB bao gồm 372451 dữ liệu hình ảnh. Với mỗi 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.

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 ký tự. Ở đây ta sử dụng chính tập đặc trưng trích chọn mà ta đã huấn luyện trong bước trước.

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

3.3.1. Lưu đồ thuật toán



Hình 3.10: Lưu đồ thuật toán nhận dạng biến kiểm soát từ video

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

Việc nhận diện biên 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 biên số.

Về cơ bản các bước xử lý và nhận diện biên kiểm soát trên từng khung hình chính là quá trình nhận diện biên 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 biên 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 biên 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 biên số. Nếu khoảng cách giữa 2 biên số lớn hơn một ngưỡng do ta quy định thì có thể kết luận đây là nhiều hoặc là một biên 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 biên số của một phương tiện, ta sẽ lưu giá trị biên số lại.

Ta cần cài đặt số khung hình mẫu chính là số khung hình cần đọc liên tiếp để xác định biên số của phương tiện. Số lượng mẫu lấy biên 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 biên số tương ứng với từng khung hình

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

Bởi các nhãn biến 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	90A12357
----------	----------	----------	----------	----------

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

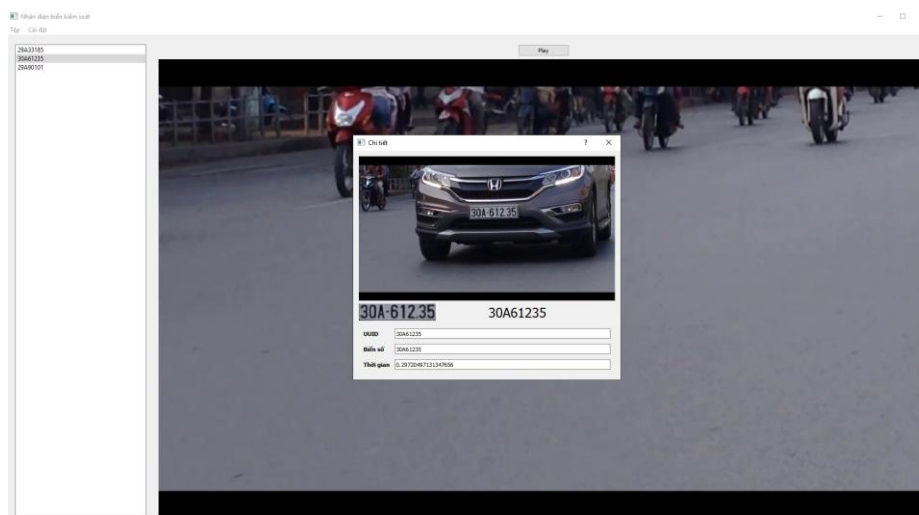
- Vị trí đầu tiên tất cả đều là 9 \Rightarrow 9
- Vị trí thứ 2 tất cả đều là 0 \Rightarrow 0
- Vị trí thứ 3 tất cả đều là A \Rightarrow A
- Vị trí thứ 4 có tần suất số 1 xuất hiện lớn nhất \Rightarrow 1
- Vị trí thứ 5 có tần suất số 2 xuất hiện lớn nhất \Rightarrow 2
- Vị trí thứ 6 có tần suất số 2 xuất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

\Rightarrow 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 cách 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 3.11: 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 biến 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:

Thời gian	Video	Kết quả
Buổi sáng	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
Buổi tối	1920x1080 / 60fps	7/10
	1366x768 / 50fps	7/10
	800x600 / 30fps	5/10
	600x400 / 20fps	3/10

- 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 biếm kiểm soát phương tiện giao thông
- Nghiên cứu các vấn đề về xử lý, nhận dạng ảnh
- Xây dựng thuật toán nhận diện biếm kiểm 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 biếm 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. Dollár 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.