|  |
| --- |
| HỌC VIỆN KỸ THUẬT MẬT MÃ  **KHOA CÔNG NGHỆ THÔNG TIN**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| **BÁO CÁO MÔN HỌC**  **THỰC TẬP CƠ SỞ CHUYÊN NGÀNH** |
| **Đề tài:**  **Phát triển xe tự hành dựa trên học máy**  **Sinh viên thực hiện:**  Đỗ Võ Hữu Tùng - CT040153  Nguyễn Xuân Văn - CT040154  Nhóm 52  **Giảng viên hướng dẫn:**  TS. Phạm Văn Hưởng |
| Hà Nội, 2022 |

MỤC LỤC

[DANH MỤC HÌNH ẢNH i](#_Toc126743277)

[LỜI MỞ ĐẦU ii](#_Toc126743278)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT XỬ LÝ ẢNH VÀ NHẬN DẠNG 1](#_Toc126743279)

[1.1 Tổng quan về xử lý ảnh 1](#_Toc126743280)

[1.1.1 Khái niệm về xử lý ảnh 1](#_Toc126743281)

[1.1.2 Các bước cơ bản trong xử lý ảnh 1](#_Toc126743282)

[1.2 Một số khái niệm cơ bản về ảnh 2](#_Toc126743283)

[1.2.1 Ảnh và điểm ảnh 2](#_Toc126743284)

[1.2.2 Mức xám 2](#_Toc126743285)

[1.2.3 Độ phân giải của ảnh 2](#_Toc126743286)

[1.2.4 Ngưỡng 3](#_Toc126743287)

[1.3 Các bài toán thường gặp trong xử lý ảnh 4](#_Toc126743288)

[1.3.1 Khử nhiễu 4](#_Toc126743289)

[1.3.2 Trích xuất đặc trưng 4](#_Toc126743290)

[1.3.3 Nhận dạng 4](#_Toc126743291)

[CHƯƠNG 2: ỨNG DỤNG XỬ LÝ ẢNH VÀ THUẬT TOÁN SUPPORT VECTOR MACHINE TRONG BÀI TOÁN PHÁT HIỆN VÀ PHÂN LỚP BIỂN BÁO GIAO THÔNG ĐƯỜNG BỘ 6](#_Toc126743292)

[2.1 Tổng quan về hệ thống phát hiện và phân lớp biển báo giao thông 6](#_Toc126743293)

[2.2 Phát hiện biển báo bằng thuật toán xử lý ảnh 7](#_Toc126743294)

[2.2.1 Phân đoạn ảnh 7](#_Toc126743295)

[2.2.2 Phát hiện vùng ứng viên 8](#_Toc126743296)

[2.3 Trích xuất đặc trưng HOG 8](#_Toc126743297)

[2.3.1 Tổng quan về đặc trưng HOG 8](#_Toc126743298)

[2.3.2 Quy trình trích xuất đặc trung HOG trên ảnh 9](#_Toc126743299)

[2.4 Thuật toán Support Vector Machine (SVM) 11](#_Toc126743300)

[2.4.1 Siêu phẳng (hyperplane) 11](#_Toc126743301)

[2.4.2 Khoảng các từ một điểm đến siêu phẳng 12](#_Toc126743302)

[2.4.3 Siêu phẳng tối ưu 13](#_Toc126743303)

[2.4.4 Giải quyết vấn đề tối ưu hóa 18](#_Toc126743304)

[CHƯƠNG 3: THỰC NGHIỆM 22](#_Toc126743305)

[3.1 Xử lý hình ảnh – phát hiện làn đường 22](#_Toc126743306)

[3.2 Xử lý hình ảnh – nhận diện biển báo 24](#_Toc126743307)

[3.2.1 Huấn luyện mô hình học máy 24](#_Toc126743308)

[3.2.2 Nhận diện biển báo 27](#_Toc126743309)

[3.3 Sản phẩm thực tế 32](#_Toc126743310)

[KẾT LUẬN 33](#_Toc126743311)

[TÀI LIỆU THAM KHẢO 34](#_Toc126743312)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Các mục đích chính của xử lý ảnh 1](#_Toc126743313)

[Hình 1.2 Các giai đoạn cơ bản của chương trình xử lý ảnh 1](#_Toc126743314)

[Hình 1.3 Ảnh gốc (trái) và ảnh phân ngưỡng (phải) 3](#_Toc126743315)

[Hình 2.1 Quy trình tổng quát của quá trình phát hiện và phân loại biển báo 6](#_Toc126743316)

[Hình 2.2 Ảnh RGB và ảnh HSV 7](#_Toc126743317)

[Hình 2.3 Ảnh đen - trắng 7](#_Toc126743318)

[Hình 2.4 Phát hiện vùng ứng viên dựa vào tọa độ. 8](#_Toc126743319)

[Hình 2.5 Dữ liệu linearly và non linearly separable 13](#_Toc126743320)

[Hình 2.6 Hai siêu phẳng thỏa mãn rằng buộc. 15](#_Toc126743321)

[Hình 2.7 z0 là một điểm thuộc H1 16](#_Toc126743322)

[Hình 3.1 Hình ảnh mô phỏng một đoạn rẽ 23](#_Toc126743323)

[Hình 3.2 Mô phỏng 3 tình huống 23](#_Toc126743324)

[Hình 3.3 Vấn đề của phương pháp 24](#_Toc126743325)

[Hình 3.4 Nhóm các biển báo sử dụng làm tập huấn luyện 25](#_Toc126743326)

[Hình 3.5 Ảnh gốc 26](#_Toc126743327)

[Hình 3.6 Ảnh trích xuất đặc trưng HOG 26](#_Toc126743328)

[Hình 3.7 Ảnh sau khi cải thiện độ tương phản 28](#_Toc126743329)

[Hình 3.8 Ảnh sau khi cải thiện 28](#_Toc126743330)

[Hình 3.9 Ảnh sau khi nhị phân hóa 29](#_Toc126743331)

[Hình 3.10 Ảnh sau khi lọc màu 30](#_Toc126743332)

[Hình 3.11 Ảnh quay bằng camera sau khi tiền xử lý 30](#_Toc126743333)

[Hình 3.12 Kết quả nhận dạng biển báo 31](#_Toc126743334)

[Hình 3.13 Mô hình xe tự hành 32](#_Toc126743335)

LỜI MỞ ĐẦU

Trong những năm gần đây trên thế giới, đi cùng với sự phát triển cùa quy mô sản xuất là sự bùng nổ trong nghiên cứu và phát triển các hệ thống robot di động tự trị (autonomous mobile robot) có khả năng ứng dụng được vào những công việc khác nhau trong kho bãi, nhà xưởng. Robot di động là một máy tự động có khả năng di chuyển đến đích trong một môi trường nào đó. Đây là điều khác với các robot công nghiệp (như tay máy có khớp quay và đầu kẹp) được gắn với một không gian cố định. Khởi đầu bằng các xe vận tải tự động AGV (autonomous guided vehicles) đã được ứng dụng rất thành công trong công nghiệp, các robot di động thông minh hiện nay vẫn đang trong thời kỳ phát triển và được coi là có tiềm năng rất lớn trong tương lai gần. Các tiến bộ công nghệ mới trong việc phát triển các thiết bị cảm biến (sensor) và khả năng tính toán cùa các hệ xử lý đã thúc đẩy mức độ tự trị trong sự vận hành các robot di động. Mặt khác, ứng dụng của robot di động trong các môi trường khác như dịch vụ, giãi trí, y tế, an ninh, quân sự cũng hứa hẹn có những tiến bộ quan trọng trong quá trình thiết kế và phát triển các hệ thống này.

Không kể hoạt động của các bộ phận gắn trên robot di động, bài toán dẫn đường cho sự di chuyển của robot từ một điểm xuất phát tới đích một cách an toàn, được gọi tắt là “điều hướng cho robot di động”, là bài toán chính yếu trong các nghiên cứu về robot di động hiện nay. Không khác nhiều so với hành vi của con người trong việc điều hướng, muốn giải quyết được bài toán này thì robot phải tự xác định được vị trí cùa mình trong môi trường (positioning), xác lập được bản đồ môi trường nếu cần thiết (mapping), vạch ra được quỹ đạo đi tới đích (path planning) và xuất ra cách thức điều khiển đi trên quỹ đạo (path control) và tránh vật cản (obstacle avoidance) trên đường đi. Để làm được điều đó, một hệ thống điều hướng của robot di động thông thường có thê chia thành 4 khối [1]: hệ thống cảm biến, hệ thống nhận thức, hệ thống lập kế hoạch di chuyển và hệ thống điều khiển động cơ. Robot di động cảm nhận về môi trường bên ngoài bằng nhiều cảm biến khác nhau gắn trên nó. Dữ liệu từ cảm biến được hệ thống nhận thức xử lý, kết hợp thành các thông tin có ý nghĩa. Hệ thống lập kế hoạch di chuyển sẽ sử dụng những thông tin đó để đưa ra đường đi tối ưu đến vị trí đích. Cuối cùng, cơ cấu chấp hành là hệ thống động cơ, chịu trách nhiệm đưa chiếc xe di chuyển theo đúng kế hoạch đã lập ra..

Xe tự hành là một trường hợp đặc biệt của robot tự hành, khi mà môi trường hoạt động của nó không chỉ gói gọn ờ khu vực nhà xưởng (in-door). Phát triển xe tự hành theo hướng đã nêu có thể giúp chiếc xe hoạt động chính xác ở các môi trường đó, nơi có rất nhiều vách ngăn, tường bao để các cảm biến có thể thu thập thông tin. Tuy nhiên, trong môi trường đường phố thực tế (out-door), chiếc xe cần tuân thủ theo các chỉ dẫn làn đường và biển báo giao thông, cần nhận biết các đối tượng giao thông khác như xe cộ, người đi đường một cách chính xác ở khoảng cách an toàn. Do vậy phương pháp kết hợp cảm biến cho các công việc này trờ nên không khả thi. Đây chính là lý do khiến dữ liệu hình ảnh trở thành một nguồn thông tin vô cùng hữu ích để chiếc xe có thể thực sự quan sát về môi trường cùa mình [2] [3].

Chính vì thế chúng em quyết định chọn để tài “**Phát triển xe tự hành dựa trên học máy”** nghiên cứu về lĩnh vự xử lý ảnh, ứng dụng các phương pháp xử lý ảnh kết hợp cùng mô hình phân lớp SVM (Support Vector Machine) trong công việc phát hiện và nhận dạng biển báo giao thông đường bộ. Triển khai lập trình hệ thống thử nghiệm dựa trên nền tảng ngôn ngữ lập trình Python và các thư viện hỗ trợ liên qua.

Trong quá trình làm báo cáo, do kiến thức còn hạn chế và thiếu nhiều kinh nghiệm nên không thể tránh khỏi những sai sót, chúng em rất mong nhận được sự đóng góp của các thầy, cô và các bạn để báo cáo được hoàn thiện hơn.

**Nhóm sinh viên thực hiện**

# CƠ SỞ LÝ THUYẾT XỬ LÝ ẢNH VÀ NHẬN DẠNG

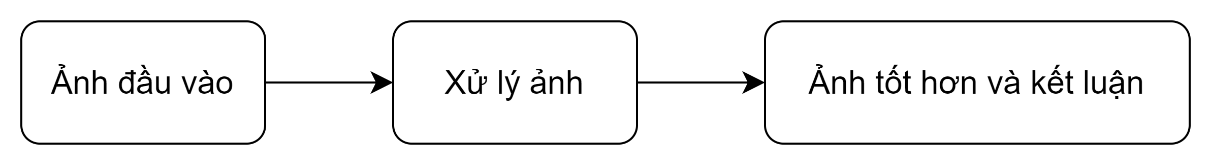
## Tổng quan về xử lý ảnh

### Khái niệm về xử lý ảnh

Xử lý ảnh là một trong những mảng nghiên cứu quan trọng nhất trong lĩnh vự thị giác máy tính (Computer Vision), là tiền đề cho nhiều ứng dụng mới thuộc lĩnh vực này. Hai nhiệm vụ cơ bản của quá trình xử lý ảnh là nâng cao chất lượng thông tin hình ảnh va xử lý số liệu cung cấp cho các quá trình khác trong đó có việc ứng dụng thị giác máy tính vào điều khiển.

Con người thu nhận thông tin qua các giác quan trong đó thị giác đóng vai trò quan trọng nhất. Sự phát triển nhanh của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển mạnh mẽ và ngày càng có nhiều ứng dụng trong cuộc sống. Xử lý ảnh đóng một vai trò quan trọng trong tương tác người - máy.

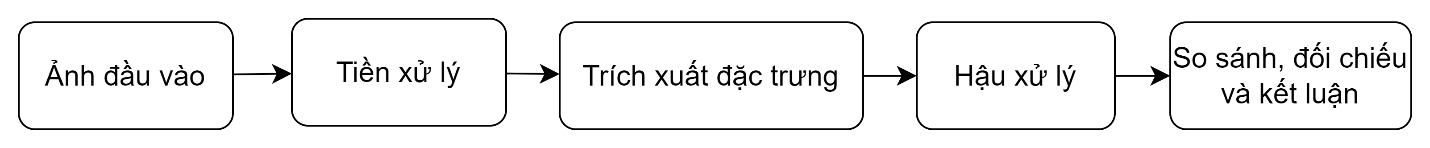
Quá trình xử lý nhận dạng ảnh là một quá trình thao tác nhằm biến đổi một ảnh đầu vào để cho ra một kết quả mong muốn kết quả đầu ra của quá trình xử lý ảnh có thể là một ảnh chất lượng tốt hơn hoặc một kết luận.



Hình 1.1 Các mục đích chính của xử lý ảnh

### Các bước cơ bản trong xử lý ảnh

Quá trình xử lý một ảnh đầu vào nhằm thu được một ảnh đầu ra mong muốn thường phải trải qua rất nhiều bước khác nhau. Các bước cơ bản của một quá trình xử lý ảnh được thể hiện thông qua hình sau:



Hình 1.2 Các giai đoạn cơ bản của chương trình xử lý ảnh

Dựa vào sơ đồ trên ta có thể thấy quá trình xử lý ảnh thông qua 5 bước chính:

* Thu thập ảnh đầu vào
* Tiền xử lý
* Trích xuất đặc trưng
* Hậu xử lý
* So sánh, đối chiếu và kết luận

## Một số khái niệm cơ bản về ảnh

### Ảnh và điểm ảnh

Điểm ảnh (pixel) được xem như là dấu hiệu hay cường độ sáng tại một tạo đột trong không gian của đối tượng.

Ảnh được xem như là một tập hợp các điểm ảnh, có thể coi ảnh là mảng số thực 2 chiều Im\*n, có kích thước m\*n bao gồm m\*n điểm ảnh, trong đó mỗi điểm ảnh có một giá trị xác định biều thị mức xám tại vị trí m\*n tương ứng.

### Mức xám

Một điểm ảnh có hai đặc trung cơ bản là vị trí (x, y) của điểm ảnh và độ xám của nó. Dưới đây chúng ta xem xét một số khái niệm và thuật ngữ thường dung xử lý ảnh.

Mức xám của điểm ảnh là cường độ sáng của nó được gán bắng giá trị số tại điểm đó. Các thang giá trị mức xám thông thường: 16, 32, 64, 128, 256 (mức 256 là mức phổ dụng. Lý do là kỹ thuật máy tính dùng 1 byte (8 bit) để biểu diễn mức xám. Mức xám dùng 1 byte biểu diễn: 28 = 256 mức, tức là từ 0-255).

### Độ phân giải của ảnh

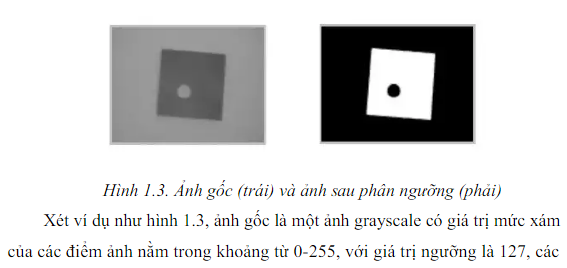
Độ phân giải của ảnh là mật độ điểm ảnh được ấn định trên ảnh được hiển thị. Khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

### Ngưỡng

Ngưỡng là một khái niệm quen thuộc trong xử lý ảnh cũng như rất nhiều giải thuật khác. Ngưỡng được dùng khá hiệu quả trọng việc nhận dạng và tìm biên ảnh. Nó dùng để chỉ một giá trị mà người ta dựa vào để phân hoạch một tập hợp thành các miền phân biệt. Trong xử lý ảnh các vùng có giá trị thỏa mãn ngưỡng thường được quy về một giá trị hay nói cách khác là giá trị mức xám tại điểm xét thỏa mãn sẽ được phân hoạch vào vùng còn lại.

Giá trị ngưỡng thường được xác định dựa vào những điểm đặc biệt hoặc dựa vào kinh nhiệm khỏa sát. Nếu dựa vào số lượng ngưỡng áp dụng cho cùng một tập dữ liệu người ta sẽ phân ra các phương pháp ứng dụng ngưỡng đơn, ngưỡng kép, hay đa ngưỡng.

Nếu dựa vào sự biến thiên của giá trị ngưỡng, trong cùng phạm vi ứng dụng người ta sẽ phân ra các phương pháp dùng ngưỡng cố định và không cố định. Ngưỡng không cố định nghĩa là giá trị của nó sẽ thay đổi tùy theo sự biến thiên của tập dữ liệu theo không gian và thời gian. Thông thường giá trị này được xác định thông qua khảo sát tập dữ liệu bằng phương pháp thống kê.



Hình 1.3 Ảnh gốc (trái) và ảnh phân ngưỡng (phải)

Xét ví dụ như hình 1.3, ảnh gốc là một ảnh grayscale có giá trị mức xám của các điểm ảnh nằm trong khoảng từ 0-255, với giá trị ngưỡng là 127, các điểm ảnh nào có giá trị lớn hơn ngưỡng ta cho nó thành màu đen (0), ngược lại là trắng (1) và cho ảnh kết quả.

## Các bài toán thường gặp trong xử lý ảnh

### Khử nhiễu

Có 2 loại nhiều cơ bản trong quá trình thu nhận ảnh:

* Nhiễu hệ thống: Nhiễn có quy luật và có thể khử bằng các phép biến đổi.
* Nhiễu ngẫu nhiên: nhiễu không rõ nguyên nhân, điều này được khắc phục bằng các phép lọc.

### Trích xuất đặc trưng

Các đặc điểm của đối tượng được trích xuất tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

* Đặc điểm không gian: Phân bổ mức xám, phân bổ xác suất, phân bổ biên độ.
* Đặc điểm biến đổi: Các đặc điểm loại này được trích xuất bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là mặt nạ đặc trưng (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v.v..).
* Đặc điểm biên và đường biên: đặc trung cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích xuất các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích xuất nhờ toán từ gradient, toán tử la bàn, toán tử Laplace, toán tử “chéo không” (zero crossing),…
* Việc trích xuất chọn hiệu quả các đặc điểm giúp chọn nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và giảm dung lượng bộ nhớ lưu trữ.

### Nhận dạng

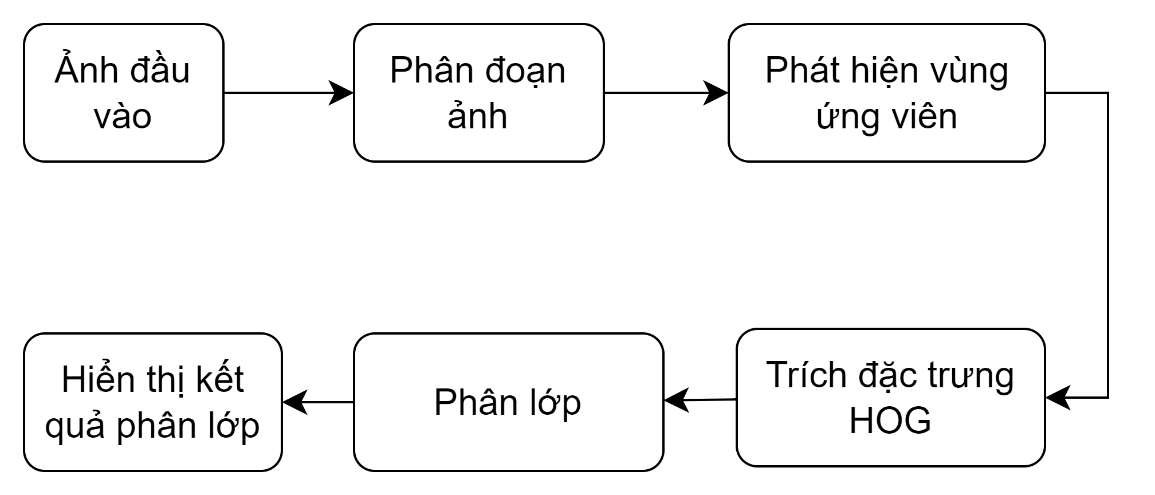
Nhận dạng tự động mô tả đối tượng, phân loại và phân nhóm các mẫu là nhưng vẫn đề quan trọng trong thị giác máy tính, được ứng dụng trong nhiều ngành khoa học khác nhau. Quá trình nhận dạng thường đi kèm với mẫu. Mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó, ta có thể làm theo hai hướng:

* Phân loại có mẫu: chẳng hạn phân tích phân biệt trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định, khi đó ta chỉ cần so sánh đối tượng đầu vào với mẫu để đưa ra kết luận.
* Phân loại không có mẫu: trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh. Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là cần nhiều những đặc trưng cần thiết, không thể do chuyên gia đề xuất mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

# ỨNG DỤNG XỬ LÝ ẢNH VÀ THUẬT TOÁN SUPPORT VECTOR MACHINE TRONG BÀI TOÁN PHÁT HIỆN VÀ PHÂN LỚP BIỂN BÁO GIAO THÔNG ĐƯỜNG BỘ

## Tổng quan về hệ thống phát hiện và phân lớp biển báo giao thông

Quy trình xử lý tổng quá của phương pháp được trình bày như trong hình 2.1.



Hình 2.1 Quy trình tổng quát của quá trình phát hiện và phân loại biển báo

Đầu tiên, ảnh đầu vào thu được từ camera sẽ được áp dụng các phương pháp phân đoạn để loại bỏ nền, làm nổi bật các đối tượng có thể là biển báo. Tiếp theo, biên của các đối tượng này sẽ đượng xử lý lọc kích thước và phân tích hình dáng để chọn ra các vùng ứng viên có thể là biển báo giao thông.

Sau đó, các vùng ứng viên sẽ được trích đặc trưng HOG, các đặc trưng này sẽ đượng đưa ra qua mô hình SVM đã được huấn luyện trước đó để nhận dạng, đưa ra kết quả cuối cùng về phân lớp của biển báo này.

Nhìn chung, hệ thống có thể chia làm hai quá trình lớn:

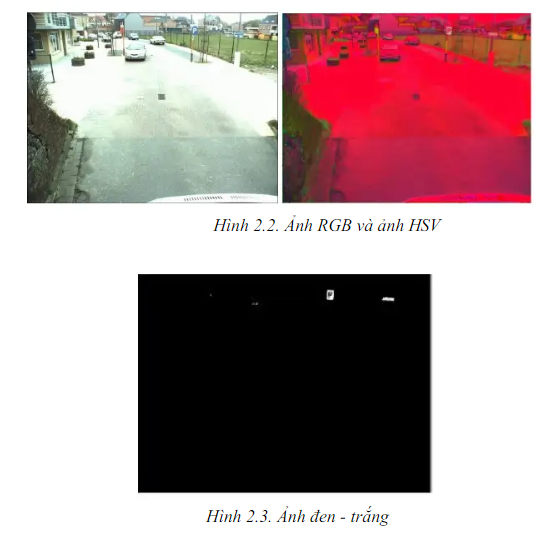
* Phát hiện biển báo trong khung hình sử dụng các thuật toán xử lý ảnh.
* Phân lớp biển báo phát hiện được bằng mô hình SVM.

## Phát hiện biển báo bằng thuật toán xử lý ảnh

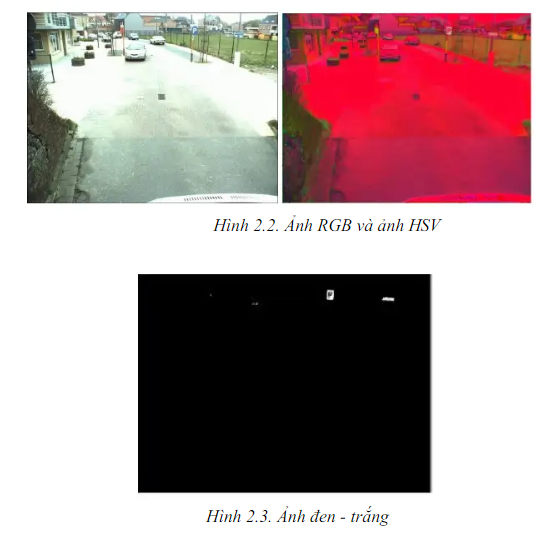
### Phân đoạn ảnh

Phân đoạn ảnh là một trong những bước chính yếu của quá trình phát hiện và phân loại biển báo giao thông, với rất nhiều tham số khác nhau cần tinh chỉnh và chọn lựa.

Hình ảnh kết quả của quá trình cân bằng histogram sẽ được chuyển đổi từ không gian màu RGB sang không gian màu HSV. Sau khi chuyển ảnh sang không gian màu HSV, chúng em xác định các giới hạn trên và dưới của các mặt nạ (mask) trong không gian màu này để lọc ra các vùng ảnh có xuất hiện màu đỏ và màu xanh. Thông số chi tiết của các mặt nạ được thể hiện dưới dạng véc-tơ [H, S, V] được biểu diễn như hình dưới.



Hình 2.2 Ảnh RGB và ảnh HSV



Hình 2.3 Ảnh đen - trắng

### Phát hiện vùng ứng viên

Hình ảnh kết quả từ quá trình xử lý trên được cung cấp cho hàm findContours() của thư viện OpenCV để dò tìm các đường viện (contour) xuất hiện trong ảnh. Mặc dù kích thước của vùng ứng viên biển báo giao thông thưc sự có tỷ lệ w/h (chiều rộng/chiều cao) xấp xỉ 1 nhưng khi truy xuất các khung hình từ camera của xe, tỉ lệ này sẽ thay đổi tùy theo khoảng cách, thời gian và góc nhìn. Tùy theo góc nhìn mà có thể chiều cao sẽ lớn hơn chiều rộng và ngược lại. Thực nghiệm cho thấy tỷ lệ phù hợp cho các vùng ứng viên với các góc nhìn khác nhau thỏa mãn w/h < 1/3 và h/w < 1/7. Vì vậy, sau khi trích được các vùng ứng viên, chúng em sử dụng ràng buộc tỷ lệ chiều rộng w và chiều cao h thỏa w/h < 1/3 và h/w < 1/7 để chọn các vùng ứng viên thực sự.



Hình 2.4 Phát hiện vùng ứng viên dựa vào tọa độ.

## Trích xuất đặc trưng HOG

### Tổng quan về đặc trưng HOG

HOG viết tắt của Histogram of Oriented Gradient là một phương pháp giúp trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.

Bản chất của phương phát HOG là sử dụng thông tin về sự phân bổ của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directions) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng được gọi là cells và với mỗi cell ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu quả nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Đặc trưng HOG cho phép miêu tả tốt các loại biển báo giao thông có hình dạng khác nhau. Vì vậy chúng em sử dụng đặc trưng HOG để trích xuất đặc trung của biển báo trên vùng ứng viên đã được tìm ra ở phần trước để phục vụ cho thao tác nhận dạng.

### Quy trình trích xuất đặc trung HOG trên ảnh

**Bước 1:** Tính cường độ và hướng biến thiên tại mỗi pixel theo công thức (2.4), (2.5).

|  |  |
| --- | --- |
| Cường độ:  Hướng: | (2.4)  (2.5) |

Trong đó,  và  là giá trị pixel (x, y).

**Bước 2:** Phân chia ảnh

Chia ảnh đầu ra ở bước trên thành nhiều khối (block), mối khối có số cell bằng nhau, mỗi cell có số pixels bằng nhau. Số khối được tính bằng công thức (2.6). Trong đó: Wimage, Himage, Wblock, Hblock, Wcell, Hcell lần lượt là chiều rộng, chiều cao của ảnh, khối và ô.

 (2.6)

**Bước 3:** Tính vector đặc trung cho từng khối

Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

* Chia không gian hướng thành p bin (số chiều vector đặc trưng của ô).
* Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin với độ lớn .

Trường hợp rời rạc hóa unsigned-HOG với p=9:

|  |  |
| --- | --- |
|  | (2.7) |

Trường hợp rời rạc hóa signed-HOG với p=18:

|  |  |
| --- | --- |
|  | (2.8) |

Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối. Số chiều vector đặc trưng khối tính theo công thức:

|  |  |
| --- | --- |
|  | (2.9) |

Trong đó: n là số ô trong khối và size feature/cell là số chiều vector đặc trưng của ô bằng 9 (unsigned-HOG) hoặc 18 (signed-HOG).

**Bước 4:** Chuẩn hóa khối

Để tăng cường hiêu năng nhận dạng, các histogram cục bộ sẽ được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một khối và sử dụng giá trị đó để chuẩn hóa tất cả các ô trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Có nhiều phương pháp có thể được dùng để chuẩn hóa khối. v là vector đặc trưng ban đầu của khối, vk  là k-norm của v (k = 1, 2), e là hằng số nhỏ. Khi đó, các giá trị chuẩn hóa có thể tính bằng một trong các công thức sau:

|  |  |  |
| --- | --- | --- |
| L2-norm: | (2.10) | |
| L1-norm: | | (2.11) |

|  |  |
| --- | --- |
| L1-sqrt: | (2.12) |

Ghép các vector đặc trưng khối sẽ thu được vector đặc trưng HOG cho ảnh. Số chiều vector đặc trưng ảnh tính theo công thức với n là số khối của ảnh, sizeblock là số chiều của vector đặc trưng khối:

|  |  |
| --- | --- |
|  | (2.13) |

## Thuật toán Support Vector Machine (SVM)

Support Vector Machine (SVM) là phương pháp phân tích lớp dựa trên lý thuyết thống kê của Vapnik và Alexei Chervonenkis xây dựng vào năm 1960 và dược sử dụng nhiều trong các ứng dụng nhận dạng chữ viết tay, nhân dạng khuôn mặt, phân loại tài liệu, tin sinh học… So với các phương pháp phân loại khác, khả năng phân loại của SVM là tương đương hoặc tốt hơn đáng kể [5]. Trong dự án này, nhóm chúng em sẽ sử dụng phương pháp phân loại SVM để phân loại biển báo giao thông.

### Siêu phẳng (hyperplane)

Chúng ta có thể hiểu siêu phẳng là gì thông qua việc nghiên cứu khái niệm đường thẳng.

Trong không gian 2 chiều, phương trình đường thẳng có dạng:

Nếu ta thay *y = x2*và *x = x1*, phương trình trở thành:

Tương đương với:

Giả sử vector **x** = (*x1, x2*) và **w** = (*a, -1*), tích vô hướng **w** và **x** là

Thay vào phương trình ta được:

Phương trình trình trên là phương trình của một siêu phẳng. Từ phương trình này, chúng ta có thể có một cái nhìn sâu sắc khác về siêu phẳng là gì: nó là tập hợp các điểm thỏa mãn .

Chúng ta có thể suy ra phương trình siêu phẳng từ phương trình của một đường thẳng, đó là bởi vì một đường thẳng cũng là một siêu phẳng. Ta sẽ nhận thấy rằng, trên thực tế, một đường thẳng là một không gian hai chiều được bao quanh bởi một mặt phẳng có ba chiều. Tương tự, điểm và mặt phẳng cũng là siêu phẳng. Lợi ích của siêu phẳng là nó giúp ta làm việc dễ dàng hơn khi không gian có lớn hơn 2 chiều. Vector **w**  sẽ luôn là **pháp tuyến** của siêu phẳng

### Khoảng các từ một điểm đến siêu phẳng

Trong không gian 2 chiều, ta biết rằng khoảng cách từ một điểm có toạ độ (*x0, y0*) tới đường thẳng có phương trình được xác định bởi:

Trong không gian ba chiều, khoảng cách từ một điểm có toạ độ (*x0, y0, z0*) tới một mặt phẳng có phương trình được xác định bởi:

Việc này có thể được tổng quát lên không gian nhiều chiều: Khoảng cách từ một điểm (vector) có toạ độ x0 tời siêu phẳng có phương trình được xác định bởi:

Với với d là số chiều của không gian.

### Siêu phẳng tối ưu

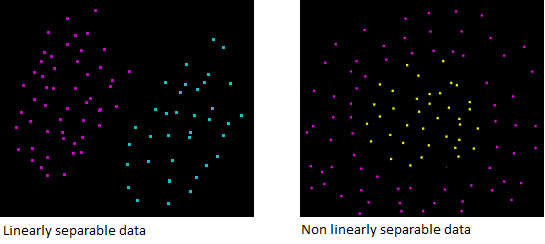
Để phân chia hai lớp dữ liệu, rõ ràng là có rất nhiều siêu phẳng có thể làm được điều này. Mặc dù vậy, mục tiêu của chúng ta là tìm ra siêu phẳng có lề rộng nhất tức là có khoảng cách tới các điểm của hai lớp là lớn nhất.

Điều này có thể được thức hiện thông qua các bước sau:

**Bước 1:** Ta có một tập dữ liệu cần phải phân loại có dạng

Với mang giá trị 1 hoặc -1, xác định lớp của điểm **xi**. Mỗi **xi** làmột vector thực p-chiều. Ta cần tìm siêu phẳng có lề lớn nhất chia các điểm yi = 1 và các điểm yi = -1.

**Bước 2:**  Chọn 2 siêu phẳng phân chia tập dữ liệu sao cho không có điểm nào nằm giữa chúng. Ta chỉ có thể làm điều này nếu tập dữ liệu là linearly separable.



Hình 2.5 Dữ liệu linearly và non linearly separable

Vì vậy, ta giả sử rằng tập dữ liệu là linearly separable.

Cho một siêu phẳng tách tập dữ liệu và thỏa mãn:

Ta chọn 2 siêu phẳng và khác có các phương trình sau:

Và

Ta thấy cách đều và . Vì hằng số không quá quan trọng nên ta đặt để cho đơn giản.

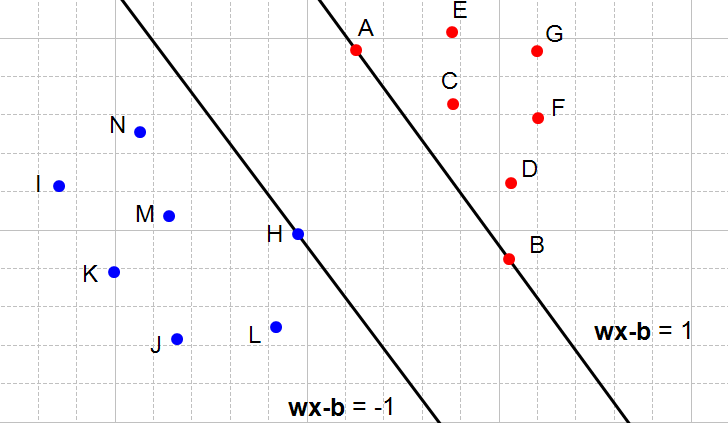
Và

Bây giờ để đảm bảo không có điểm nào nằm giữa chúng, ta sẽ chỉ lựa chọn cách siêu phẳng thỏa mãn điều kiện:

|  |  |
| --- | --- |
|  | (2.14) |

hoặc

|  |  |
| --- | --- |
|  | (2.15) |



Hình 2.6 Hai siêu phẳng thỏa mãn rằng buộc.

Bằng cách xác định các ràng buộc này, ta có thể tìm ra 2 siêu phẳng không có điểm nào nằm giữa chúng. Và điều này có thể được áp dụng với p-chiều.

Trong toán học, mọi người thích mọi thứ được diễn đạt ngắn gọn.

Các phương trình (2.14) và (2.15) có thể được kết hợp thành một ràng buộc duy nhất:

Bắt đầu với phương trình (2.15) khi ta nhân cả 2 vế với *yi*(trong trường hợp này *yi* = -1 vì các phần tử thuộc lớp -1). Phương trình (2.15) được viết thành:

|  |  |
| --- | --- |
|  | (2.16) |

Tương tự với phương trình (2.14), nhưng lúc này *yi* = 1 nên nó không làm thay đổi dấu của bất phương trình.

|  |  |
| --- | --- |
|  | (2.17) |

Kết hợp phương trình (2.16) và (2.17) ta được:

|  |  |
| --- | --- |
|  | (2.18) |

**Bước 3:** Cực đại hóa khoảng cách giữa 2 siêu phẳng

Trước khi cố gắng tối đa hóa khoảng cách giữa hai siêu phẳng, trước tiên chúng ta sẽ tìm cách nào để tính khoảng cách đó.

Giả sử:

* là siêu phẳng có phương trình:
* là siêu phẳng có phương trình:
* là một điểm thuộc siêu phẳng

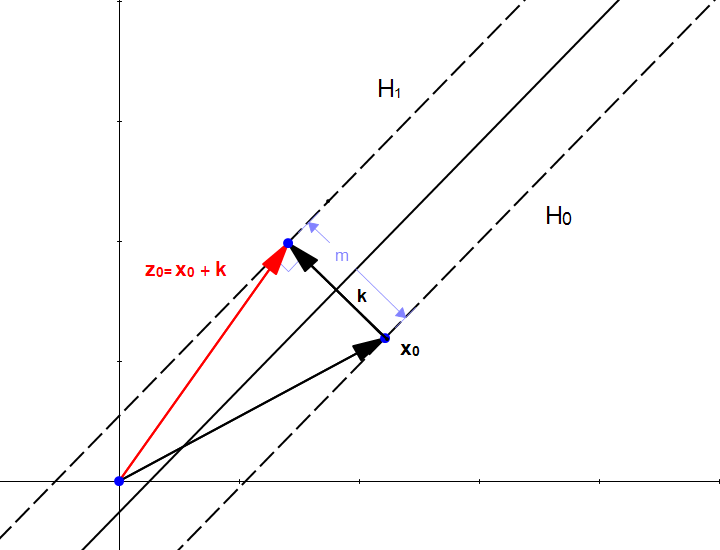
Gọi là khoảng cách vuông góc từ đến siêu phẳng .Theo định nghĩa, chính là margin (lề). Vì thuộc nên chính là khoảng cách giữa 2 siêu phẳng và **.**

Vector đơn vị của vector **w** là: . Nếu ta nhân **u** với  ta sẻ được vector **k** = với:

* (vì **)**
* **k** vuông góc với (vì **k** cùng hướng với **w** - vector pháp tuyến của siêu phẳng )

|  |  |
| --- | --- |
|  | (2.19) |

Ta có: là một điểm thuộc siêu phẳng



Hình 2.7 z0 là một điểm thuộc H1

Vì z0 thuộc  nên:

|  |  |
| --- | --- |
|  | (2.20) |

Thay :

|  |  |
| --- | --- |
|  | (2.21) |

Thay **k**  bằng phương trình (2.19):

|  |  |
| --- | --- |
|  | (2.22) |
|  | (2.23) |

Tích vô hướng của một vectơ với chính nó là bình phương của độ dài nên:

|  |  |  |
| --- | --- | --- |
|  | (2.24) | |
|  | (2.25) |
|  | (2.26) |

Vì **x**0 thuộc  nên:

|  |  |
| --- | --- |
|  | (2.27) |
|  | (2.28) |
|  | (2.29) |

Bây giờ chúng ta có một công thức để tính toán margin:

Biến duy nhất chúng ta có thể thay đổi trong công thức này là độ dài của **w**. Ta có thể thấy, độ dài càng lớn thì margin càng nhỏ. Như vậy cực đại hóa margin cũng chính là cực tiểu độ dài của **w.** Như vậy trong số tất cả các siêu phẳng có thể đáp ứng các ràng buộc, chúng ta sẽ chọn siêu phẳng có nhỏ nhất bởi vì khi đó ta sẽ thu được margin là lớn nhất.

Điều này cho chúng ta vấn đề tối ưu hóa sau:

Ta biến đổi để bài toán trở thành:

Hệ số được thêm vào để thuận tiện về sau này, khi chúng ta sẻ sử dụng bộ giải QP để giải bài toán.

### Giải quyết vấn đề tối ưu hóa

Trong phần trên chúng ta có một hàm cần tối ưu hóa là:

với hàm ràng buộc *m* là:

Ta có hàm Lagrangian như sau:

với là hệ số Lagrange.

Bây giờ ta vấn đề cần tìm là , nhưng vấn đề chỉ có thể được giải quyết khi số lượng mẫu nhỏ (theo Tyson Smith, 2004). Vì thế, thay vì giải trực tiếp bài toàn gốc người ta thường sẽ giải bài toán đối ngẫu của nó.

Để có nghiệm của bài toán nguyên hàm, ta cần giải bài toán Lagrangian sau:



Bài toán Lagrange có *m* ràng buộc bất đẳng thức thường được giải bằng các sử dụng dạng đối ngẫu của nó.

Có đạo hàm riêng của đổi với và là:

Từ phương trình trên ta tính được:

Thay vào , ta được:

Vì tức là nên:

Đây là hàm Wolfe đối ngẫu Lagrangian.

Bài toán tối ưu hóa bây giờ được gọi là bài toán đối ngẫu Wolfe:

Ưu điểm chính của bài toán đối ngẫu Wolfe so với bài toán Lagrange là hàm mục tiêu *W* bây giờ chỉ phụ thuộc duy nhất vào hệ số Lagrange.

Bởi vì chúng ta đang giải quyết các ràng buộc về bất đẳng thức, nên có một yêu cầu bổ sung là phải thỏa mãn các điều kiện Karush-Kuhn-Tucker (KKT).

Điều kiện KKT là phép thử đạo hàm cấp một để giải quyết bài toán tối ưu hóa với điều kiện là một số điều kiện về tính đều đặn được thỏa mãn. Một trong số đó là **điều kiện Slater** và đối với các SVM điều kiện này được đảm bảo. Vì bài toán nguyên hàm mà chúng ta đang cố gắng giải là một bài toán lồi nên nếu nó thỏa mãn điều kiện Slater thì *strong duality* thỏa mãn. Và nếu *strong duality* thỏa mãn thì nghiệm của bài toán chính là nghiệm của hệ điều kiện KKT.

Các điều kiện Karush-Kuhn-Tucker là:

* Stationary:
* Primal feasibility:
* Dual feasibility:
* Complementary slackness:

# THỰC NGHIỆM

## Xử lý hình ảnh – phát hiện làn đường

Bước 1: Thresholing

Vì đường cho xe chạy sử dụng giấy trắng A4 để làm nên chúng em chỉ cần sử dụng tính năng phát hiện màu để tìm đường dẫn.

Ý tưởng ở đây chỉ đơn giản là chuyển đổi hình ảnh sang không gian HSV và sau đó áp dụng một dải màu để tìm ra đường ( chọn mọi thứ trừ màu trắng)

Bước 2: Warping Lane

Vì chúng em không muốn xử lý toàn bộ hình ảnh mà camera thu thập được, vỉ em chỉ muốn RasPi biết có khúc cua trên con đường ngay tại thời điểm đó chứ không phải là vài giây phía trước nên em cần cắt hình ảnh mà camera thu thập được. Nhưng hơn thế nữa, em còn muốn PasPi nhìn con đường như thể đang nhìn từ trên xuống. Đây gọi là “bird eye view” và nó rất quan trọng vì nó sẽ cho phép RasPi dễ dàng tìm thấy đường cong hơn.

Để làm điều này trước tiên chúng em xác định các điểm ban đầu. Những điểm này thường được xác định thủ công vì thế em có thể sử dụng các trackbar để thử nghiệm các giá trị khác nhau với ý tưởng là để có được một hình chữ nhật khi đường thẳng.

Bước 3: Histogram

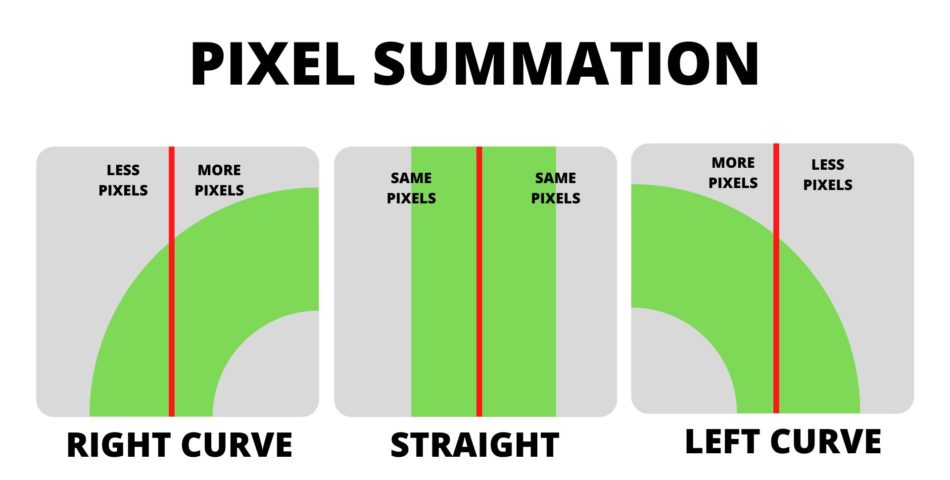
Đây là phần quan trọng nhất để tìm ra khúc cua trên đường đi. Ý tưởng là sử dụng tổng các pixel mà các ô là màu trắng theo chiều y

Chart

Description automatically generated

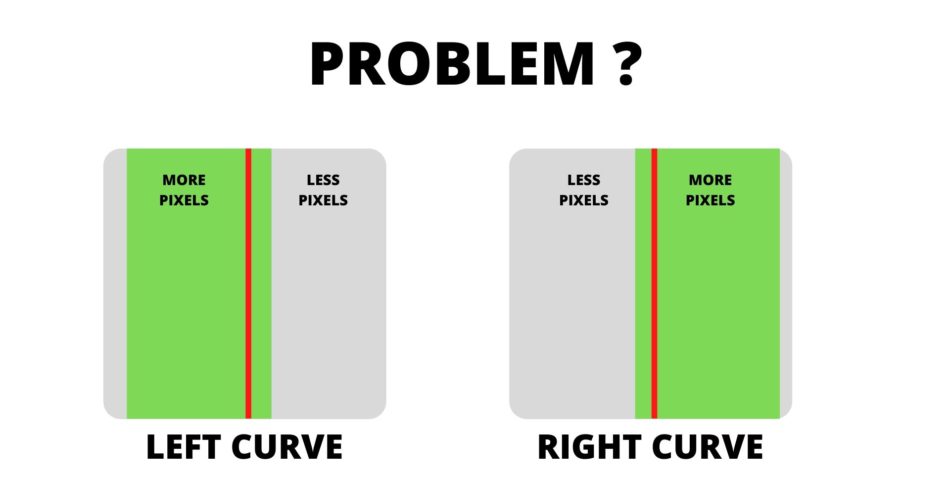
Hình 3.1 Hình ảnh mô phỏng một đoạn rẽ

Hình trên hiển thị tất cả các pixel màu trắng có giá trị 255 và tất cả các pixel màu đen có giá trị 0. Bây giờ nếu tính tổng các pixel trong cột đầu tiên thì kết quả là 1275. Cứ như thế ta tính ra được tổng pixel ở từng cột. Vì hình ảnh có kích thước 480 x 240 px nên ta có 480 giá trị. Sau khi tính tổng, ta có thể xem có bao nhiêu giá trị vượt quá ngưỡng nhất định ( do ta chọn) ví dụ như 1000. Như vậy, trong ví dụ trên ta có 8 cột bên trái và 3 cột bên phải có tổng số pixel vượt ngưỡng 1000 suy ra đường cong đang hướng về bên trái.



Hình 3.2 Mô phỏng 3 tình huống

Tuy nhiên phương pháp này có một số vấn đề:



Hình 3.3 Vấn đề của phương pháp

Ở đây, mặc dù không có đường cong nhưng vì có nhiều pixel hơn một bên, thuật toán sẽ kết luận là đường cong về bên trái hoặc phải. Vì thế để khắc phục chúng em phải điều chỉnh đường trung tâm sau đó mới so sánh các pixel ở cả hai bên.

Bước 4: Tính trung bình

Khi có giá trị đường cong, chúng em sẽ thêm nó vào danh sách có thể tính giá trị trung bình. Tính trung bình sẽ cho phép chuyển động trơn tru hơn.

## Xử lý hình ảnh – nhận diện biển báo

### Huấn luyện mô hình học máy

Bước 1: Thu nhập dữ liệu

Để triển khai và thực hiện huấn luyện học máy để nhận diện vật thể thì việc quan trọng nhất là cần cung cấp cho mô hình học máy dữ liệu hình ảnh từ thế giới thực để học và dự đoán, cụ thể trong khuôn khổ báo cáo này là các biển báo để chỉ dẫn đường đi cho xe tự hành. Dữ liệu này có thể được sử dụng để huấn luyện mô hình hoặc để kiểm tra mô hình của chúng ta đã được huấn luyện chính xác hay chưa.

Dữ liệu mà mô hình cần sử dụng sẽ là tập hình ảnh các biển báo:



Hình 3.4 Nhóm các biển báo sử dụng làm tập huấn luyện

Bước 2: Tiền xử lý hình ảnh

Xử lý độ nghiêng của ảnh: Đầu tiên chúng em sẽ kiểm tra tính đối xứng của ảnh. Nếu hình ảnh truyền vào đã "đều" và không cần phải tiến hành bất kỳ xử lý gì. Trong trường hợp này, hàm sẽ trả về bản sao của hình ảnh. Nếu không đều thì tính độ nghiêng của ảnh, sau đó thực hiện đều hóa và khử nghiêng.

Bước 3: Trích xuất đặc trưng hình ảnh

Để trích xuất đặc trưng, chúng em sẽ sử dụng HOG (Histogram of oriented gradient).

Khởi tạo bộ mô tả đặc trưng HOG với các tham số thích hợp. Sau đó, sẽ tính mô tả HOG của mỗi hình trong tập ảnh và lưu trữ lại.



Hình 3.5 Ảnh gốc



Hình 3.6 Ảnh trích xuất đặc trưng HOG

Bước 4: Thực hiện huấn luyện

Sử dụng thuật toán Support Vector Machine (SVM). Mô hình học máy sử dụng SVC để xây dựng nên mô hình SVM với kernel là RBF với 2 Hyperparameter:

* C: Tham số phạt hay mức độ chấp nhận lỗi
* Gamma: Hệ số điều chỉnh hệ số radial basis function (rbf) trong kernel. Gamma sẽ biểu thị xem mức độ xa gần của các vector hỗ trợ (support vector)

Sau đó đưa dữ liệu HOG được trích xuất và nhãn vào mô hình để thực hiện

huấn luyện

Mục đích của việc sử dùng SVC ở đây là do mục đích của chúng em là phân lớp, sử dụng SVC có thể tối ưu hóa hiệu suất và tính chính xác so với việc sử dụng SVM toàn diện.

SVM (Support Vector Machine) là một mô hình phân lớp đa nhãn hoặc hồi qui phù hợp với nhiều tình huống phân lớp. Trong khi đó, SVC (Support Vector Classification) là một trong những phần của SVM dành cho việc phân lớp.

SVC sử dụng mô hình SVM để tìm ra một hyperplane tốt nhất để phân chia dữ liệu vào các lớp tương ứng. Nó chỉ quan tâm đến những điểm dữ liệu là hạt giữa giữa các lớp và không quan tâm đến các điểm dữ liệu nằm ngoài các lớp. Điều này có nghĩa là SVC chỉ là một phần của SVM dành cho việc phân lớp và được sử dụng khi mục đích chính của bạn là phân lớp.

### Nhận diện biển báo

Bước 1: Nhận dữ liệu từ camera

Mở camera bằng thư viện Open cv và lấy ra các frame ảnh để thực hiện các bước tiếp theo.

Bước 2: Tiền xử lý hình ảnh

Cải thiện độ tương phản của ảnh. Bằng cách thực hiện việc chuyển đổi màu từ không gian màu BGR sang YCrCb và chia hình ảnh thành 3 kênh: Y, Cr, và Cb. Sau đó, nó cải thiện độ tương phản của kênh Y và trộn các kênh lại và chuyển đổi hình ảnh trở lại không gian màu BGR.



Hình 3.7 Ảnh sau khi cải thiện độ tương phản

Nâng cao biên của ảnh. Bằng cách sử dụng bộ lọc Gaussian để giảm nhiễu và làm mịn ảnh. Sau đó ảnh được chuyển đổi sang đen trắng (grayscale). Cuối cùng sử dụng thuật toán Laplacian để cải thiện biên ảnh.



Hình 3.8 Ảnh sau khi cải thiện

Nhị phân hóa ảnh bằng cách sử dụng phương pháp thresholding. Thresholding là một quá trình chuyển đổi hình ảnh thành hình ảnh nhị phân bằng cách chọn một ngưỡng và chuyển đổi tất cả các pixel có giá trị lớn hơn ngưỡng thành giá trị trắng và các pixel còn lại thành giá trị đen.



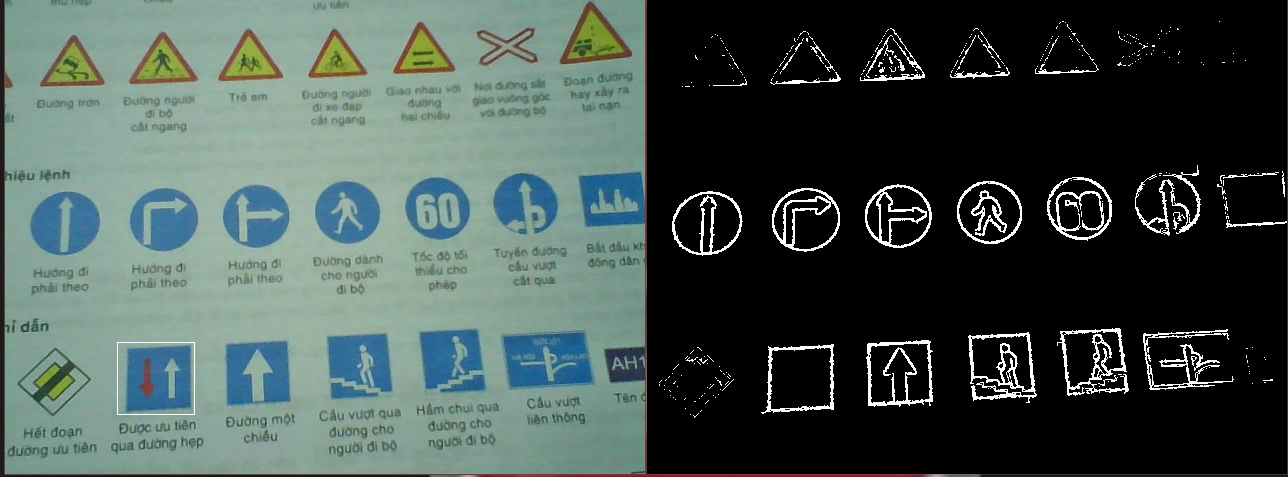
Hình 3.9 Ảnh sau khi nhị phân hóa

Loại bỏ các thành phần nhỏ. Bằng cách tìm tất cả các thành phần liên kết, sau đó thông qua kích thước các thành phần để loại bỏ các khuôn mẫu nhỏ.

Lọc màu. Đầu tiên loại bỏ nhiễu và làm mịn hình ảnh bằng Gaussian. Sau đó chuyển ảnh về HSV. Định nghĩa khoảng màu xanh, trắng và đen và tách các màu ra khỏi ảnh. Cuối cùng kết hợp các mặt nạ lại mới nhau.



Hình 3.10 Ảnh sau khi lọc màu



Hình 3.11 Ảnh quay bằng camera sau khi tiền xử lý

Bước 3: Trích xuất các thành phần biển báo khỏi ảnh

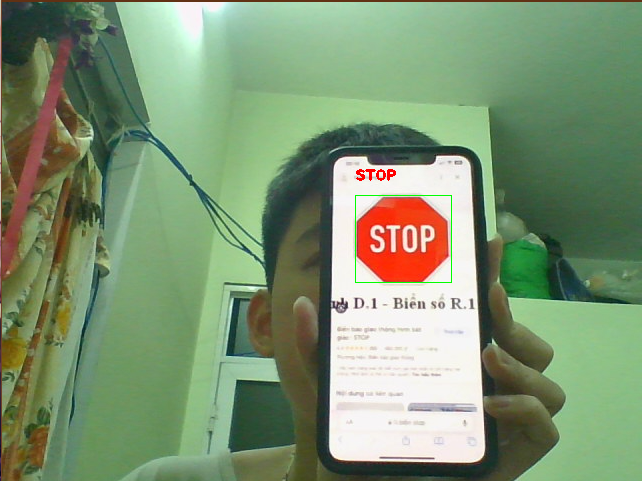
Đầu tiên, chúng em tìm các đường viền trong hình ảnh. Đường viền là các đường bao quanh các vùng có màu khác nhau trong hình. Sau đó duyệt qua tất cả các đường viền, tìm thông số về kích thước và tọa độ trung tâm của đường viền, tính khoảng cách từ mỗi điểm trên viền tới tâm của viền. Từ đó xác định được đường viền có hình dạng biển báo hay không. Cuối cùng cập nhật tọa độ biển và cắt hình biển báo ra khỏi ảnh gốc

Bước 4: Trích xuất đặc trưng ảnh

Tương tự với việc trích xuất khi huấn luyện mô hình

Bước 5: Sử dụng mô hình để nhận diện biển báo

Đưa các đặt trưng HOG từ hình ảnh vừa trích xuất được đưa vào mô hình đã được huấn luyện. Mô hình sẽ dự đoán và đưa ra nhãn của hình ảnh



Hình 3.12 Kết quả nhận dạng biển báo

## Sản phẩm thực tế

|  |  |
| --- | --- |
| Không có mô tả. | Không có mô tả. |

Hình 3.13 Mô hình xe tự hành

KẾT LUẬN

Hệ thống xe tự hành bằng học máy đã đáp ứng được yêu cầu về việc phát hiện làn đường và nhận diện chính xác biển báo để có thể xử lý các tình huống mô phỏng như ngoài thực tế. Đây là hướng nghiên cứu không còn mới ở cả Việt Nam và thế giới. Trên thế giới các hệ thống với mục đích tương tự hiện vẫn đang được nghiên cứu và thử nghiệm, đặc biệt với các trung tâm nghiên cứu của các hãng chế tạo ô tô như Toyota, Ford.

Với thời gian nghiên cứu có hạn và lượng kiến thức hạn chế nên đề tài còn nhiều nhược điểm cần phải khắc phục, tiêu biểu nhất là số lượng biển báo giao thông nhóm nhận dạng được còn ít và tỉ lệ nhận dạng đúng chưa cao.

Trong tương lai nhóm sẽ cố gắng cải thiện về hiệu năng của hệ thống để nâng cao hiệu quả nhận diện của thuật toán cũng như độ chính xác của mô hình học máy.

TÀI LIỆU THAM KHẢO

1. Takacs, Arpad & Rudas, Imre & Bosl, Dominik & Haidegger, Tamas. (2018). Highly Automated Vehicles and Self-Driving Cars [Industry Tutorial]. IEEE Robotics & Automation Magazine. 25. 106-112. 10.1109/MRA.2018.2874301.
2. Treml, M., et al.:Speeding up Semantic Segmentaion for Autonomous Driving. Proceedings of the Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 1-7 (2016).
3. Yuan, Chang & Chen, Hui & Liu, Ju & Zhu, Di & Xu, Yanyan, Robust Lane Detection for Complicated Road Environment Based on Normal Map. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2868976, 2018.
4. Nguyễn Văn Long, “Tìm hiểu và đề xuất phương pháp nhận dạng và phân loại biển báo giao thông ở Việt Nam”, Luận văn Thạc sĩ, Đại học Duy Tân, 2016, tr33-83.
5. Hà Thị Kim Duyên, Lê Mạnh Long, Nguyễn Đức Duy, Phan Sỹ Thuần, Nguyễn Ngọc Hải, Nguyễn Thị Tú Uyên, “NGHIÊN CỨU VÀ PHÁT TRIỂN HỆ THỐNG XE TỰ HÀNH ỨNG DỤNG TRÍ TUỆ NHÂN TẠO”, hội nghị HAUI tổ chức vào tháng 6/2020.
6. Hiep Do Quang, Tien Ngo Manh, Cuong Nguyen Manh, Dung Pham Tien, Manhj Tran Van, Kiem Nguyen Tien, Duy Nguyen Duc, “An Approach to Design Navigation System for Omnidirectional Mobile Robot Based on ROS”, 2020 3rd International Conference on Control, Robotics and Informations (ICCRI 2020).
7. Van Nguyen Thi Thanh, Tien Ngo Manh, Cuong Nguyen Manh, Dung Pham Tien, Manh Tran Van, Duyen Ha Thi Kim, Duy Nguyen Duc, “Autonomous Navigation for Omnidirectional Robot Based on Deep Reinforcement Learing”, 2020 3rd International Conference on Control, Robotics and Informations (ICCRI 2020).
8. Maitre, Henri. (2013). Image Processing: Overview and Perspectives. Intelligent Video Surveillance Systems. 1-12. 10.1002/9781118577851.ch1.
9. Mishra, Manish & Srivastava, Monika. (2014). A view of Artificial Neural Network. 1-3. 10.1109/ICAETR.2014.7012785.
10. O’Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.
11. Saadna, Yassmina & Behloul. Ali. (2017). An overview of traffic sign detection and classification methods. International Journal of Multimedia Information Retrieval. 6. 1-18. 10.1007/s13735-017-0129-8.