**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC QUỐC GIA TP HCM**

Danh Đức Khánh Duy – Thân Thế Tùng – Nguyễn Đức Hoan

**PHÁT HIỆN và NHẬN DIỆN BIỂN BÁO GIAO THÔNg**

HCM, thứ 7 ngày 28 tháng 12 năm 2019

Mục lục

[I. Giới thiệu 2](#_Toc29408120)

[II. Phương pháp lấy hình ảnh 4](#_Toc29408121)

[III. Phát hiện biển báo 5](#_Toc29408122)

[IV. Nhận dạng biển báo 12](#_Toc29408123)

[V. Phương pháp kết hợp các thành phần 15](#_Toc29408124)

[VI. Tổng kết: 17](#_Toc29408125)

[VII. Tài liệu tham khảo 18](#_Toc29408126)

**Mục lục hình ảnh:**

[Hình 1. Công nghệ nhận diện biển báo là công nghệ tiên tiến và cần thiết cho mọi phương tiện tham gia giao thông 2](file:///C:\Users\TungTT\Desktop\Báo-cáo-đồ-án-CK-bản-đẹp.docx#_Toc29408146)

[Hình 2. Lưu đồ giải thuật của các bài toán Nhận dạng biển báo. 3](#_Toc29408147)

[Hình 3. Hệ thống biển báo Việt Nam 5](#_Toc29408148)

[Hình 4. Ảnh mẫu để tính toán tìm thông số histogram mẫu đại diện cho mỗi nhóm biển báo. 6](#_Toc29408149)

[Hình 5. Lưu đồ giải thuật cho phương pháp Histogram 6](#_Toc29408150)

[Hình 6. Ảnh minh họa kết quả sử dụng phương pháp Histogram 7](#_Toc29408151)

[Hình 7. Sơ đồ thuật toán Phát hiện biển báo 8](#_Toc29408152)

[Hình 8. Mô tả hệ màu HSV 9](#_Toc29408153)

[Hình 9. Mô tả khả năng lọc màu của mask 9](#_Toc29408154)

[Hình 10. Hình ảnh thực tế của các mask 10](#_Toc29408155)

[Hình 11. Hình ảnh biển báo sau khi đã được Phát hiện 10](#_Toc29408156)

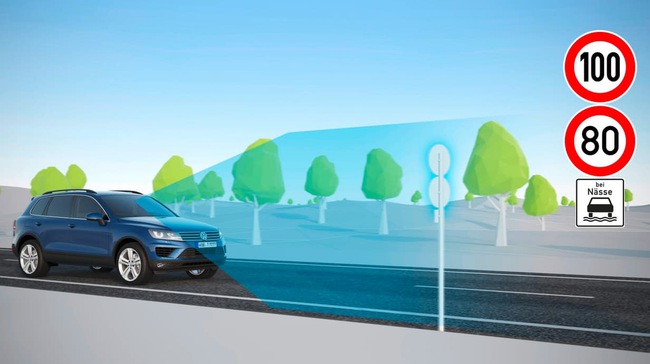
[Hình 12. Lưu đồ giải thuật Nhận dạng 13](file:///C:\Users\TungTT\Desktop\Báo-cáo-đồ-án-CK-bản-đẹp.docx#_Toc29408157)

[Hình 13. Trích xuất đặc điểm từ hình mẫu và hình ảnh từ phần Phát hiện 14](#_Toc29408158)

[Hình 14: Lưu đồ giải thuật chung của bài toán 15](file:///C:\Users\TungTT\Desktop\Báo-cáo-đồ-án-CK-bản-đẹp.docx#_Toc29408159)

[Hình 15. Kết quả sau khi Phát hiện và Nhận diện 16](#_Toc29408160)

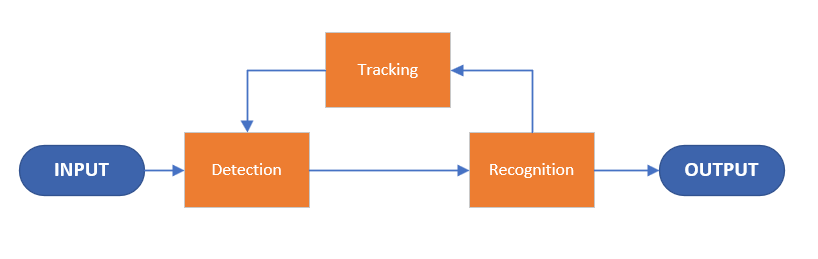
# I. Giới thiệu

Ngày nay, khi tốc độ phát triển ngày càng cao, nhu cầu tham gia giao thông của người dân ngày càng tăng và trở thành vấn đề quan trọng. Tuy nhiên, ý thức tự giác và cảnh giác của người dân chưa cao, từ những hình ảnh thực tế giao thông Việt Nam có một phần do các tài xế không quan sát và không kịp nhận ra biển báo và tín hiệu giao thông. Đây là những nguy hiểm đã được cảnh báo trước nhưng vẫn thường xuyên xảy ra, gây không ít thiệt hại về tính mạng và tài sản của người tham gia giao thông.

Hình 1. Công nghệ nhận diện biển báo là công nghệ tiên tiến và cần thiết cho mọi phương tiện tham gia giao thông

Vì vậy nhóm đã chọn đề tài “Phát hiện biển báo giao thông” để nghiên cứu với mục đích hiểu được nền tảng công nghệ, các lý thuyết căn bản trong việc xử lý hình ảnh và áp dụng trong đề tài một cách tự động.

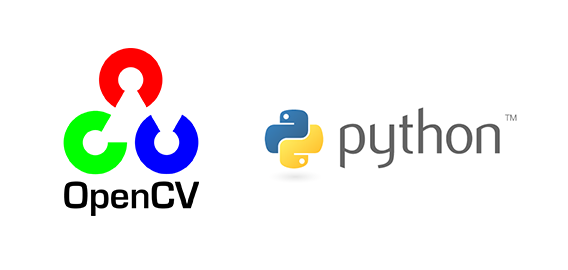
Hiện nay mỗi khi nhắc đến bài toán xử lí ảnh biển báo trong lĩnh vực thị giác máy tính, thì thường có 3 phần chính là Detection-Recognition-Tracking (Phát hiện-Nhận diện-Bám vật thể).



Hình 2. Lưu đồ giải thuật của các bài toán Nhận dạng biển báo.

Ở đồ án môn học này: nhóm sẽ tập trung vào Detection và Recognition.

* *Đối tượng, phạm vi thực hiện:* biển báo giao thông Việt Nam.
* *Phương pháp thực hiện:*
* Tìm hiểu bộ ảnh biển báo giao thông Việt Nam, thu thập hình ảnh về biển báo từ các nguồn khác nhau (vd: Google image, tự chụp, …)
* Tìm hiểu các phương pháp phát hiện biển báo dựa trên đặt trưng về màu sắc và hình dạng.
* Tìm hiểu phương pháp nhận diện biển báo dựa vào phương pháp so khớp.
* Cài đặt chương trình thử nghiệm và đánh giá kết quả nhận dạng.
* *Công cụ thực hiện:*
* Ngôn ngữ Python và các trình biên dịch hỗ trợ (vd Pycharm, VisualStudio, …).
* Bộ thư viện hỗ trợ việc xử lý ảnh Opencv phiên bản 3.4.2 và các thư viện hỗ trợ khác (vd Numpy, Math, …).

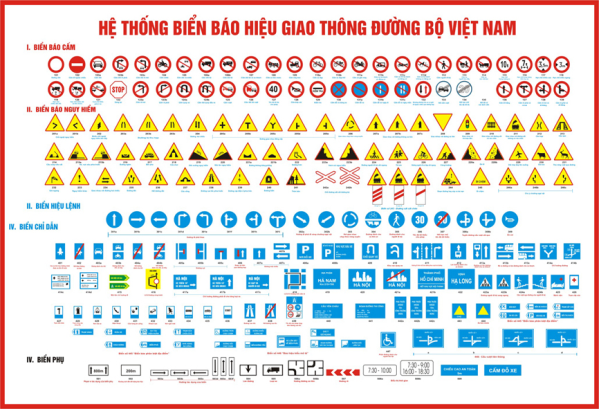


# II. Phương pháp lấy hình ảnh

* Sử dụng camera hành trình ta nhận được video trực tiếp về biển báo giao thông từ đó sử dụng các công cụ có sẵn để rút trích hình ảnh.
* Các bước lấy hình ảnh như sau:
* B1: Nhận video trực tiếp từ camera.
* B2: Lấy FpS (Fram pes Seconds) video từ đó ta lấy ra mỗi Frame để thực hiện các thuật toán.
* Nếu thuật toán hoàn thành trước khi có Frame tiếp theo, thuật toán sẽ đợi cho đến khi có Frame sẽ tiếp tục.

# III. Phát hiện biển báo

* Khi nhìn vào hệ thống biển báo Việt Nam, ta có thể dể dàng nhận ra các loại biển báo được chia thành 5 nhóm chính:
* Nhóm biển báo cấm: đa phần là viền đỏ và là hình tròn
* Nhóm biển báo nguy hiểm: có viền đỏ và là hình tam giác
* Nhóm biển hiệu lệnh: màu xanh và là hình tròn
* Nhóm biển chỉ dẫn: màu xanh và là hình chữ nhật
* Nhóm biển phụ: viền đen và là hình chữ nhật



Hình 3. Hệ thống biển báo Việt Nam

* Ban đầu nhóm thử nghiệm sử dụng phương pháp so sánh Histogram: Nhìn vào ảnh, ta có thể thấy 3 nhóm biển báo chính với 3 phổ màu khác nhau rõ rệt: Biển cấm – biển nguy hiểm – biển hiệu lệnh và chỉ dẫn tương ứng với 3 gam màu đỏ - vàng - xanh dương.



Hình 4. Ảnh mẫu để tính toán tìm thông số histogram mẫu đại diện cho mỗi nhóm biển báo.



Hình 5. Lưu đồ giải thuật cho phương pháp Histogram

* Nhóm nhận thấy đa phần các nhóm khác sau khi lấy được vùng màu binary ở bước 2 trên đây đều sẽ làm thêm bước Morphological erosion để lọc nhiễu theo như nhiều hướng dẫn trên google. Tuy nhiên nhóm không làm theo cách này vì nếu dùng giãn nở hoặc xói mòn có thể khiến cho không phát hiện được contour của ảnh biển báo cực nghiêng, vì lúc này contour rất nhỏ, nếu sử dụng morphological erosion sẽ làm mất đi contour này, khiến cho không phát hiện (detect) được vùng nghi có biển báo. Thay vào đó, để lọc đi các contour nhỏ, các contour vụn lẻ tẻ thì nhóm duyệt qua tất cả các contour, tính diện tích từng contour và chỉ xét các contour với kích thước lớn hơn mức phù hợp.

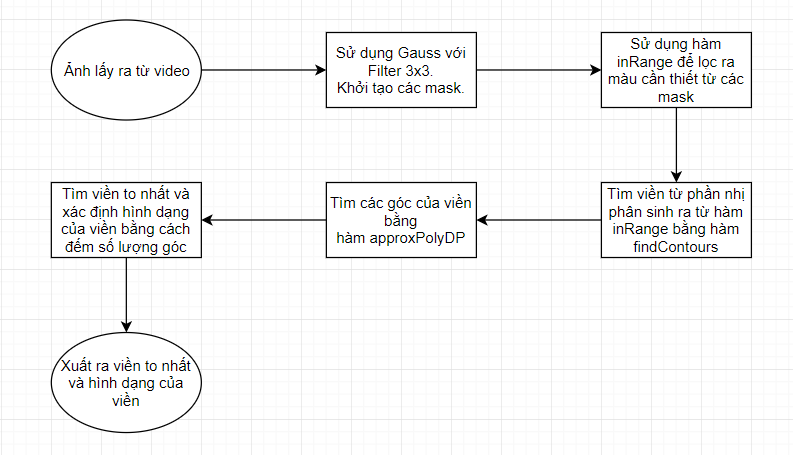


Hình 6. Ảnh minh họa kết quả sử dụng phương pháp Histogram

* Sau khi thử nghiệm phát hiện vùng có biển báo bằng phương pháp Histogram Calculation, nhóm rút ra được các ưu và nhược sau:

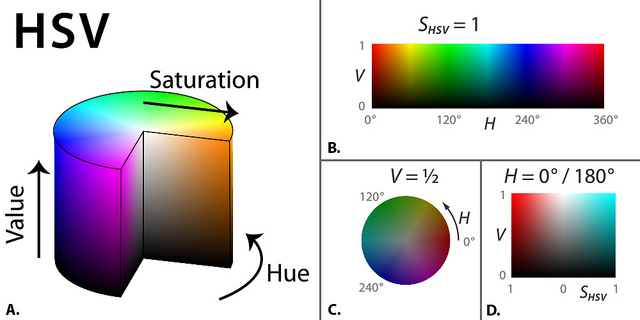
|  |  |  |
| --- | --- | --- |
| **Ưu điểm** | **Nhược điểm** | **Giải pháp khắc phục** |
| * Ảnh nghiêng góc cực lớn vẫn detect được. * Để detect biển báo đó thuộc nhóm biển báo nào (cấm, hiệu lệnh - chỉ dẫn, nguy hiểm) thì chỉ cần 1 ảnh đại diện cho mỗi nhóm biển báo. | * Màu thường bị thay đổi sắc tố phụ thuộc vào môi trường nên dễ bị nhiễu, phát hiện sai. * Biển cùng nhóm biển báo nhưng màu sắc khác (ví dụ màu tím, …) thì sẽ không detect được. * Chạy chậm vì phải tính toán histogram cho nhiều biển báo trên 1 ảnh. | * Tăng số ảnh làm template mẫu. * Kết hợp phát hiện màu sắc và phát hiện hình dạng của vật thể (hình tròn, tam giác, chữ nhật). * Thu hẹp phạm vi yêu cầu bài toán thành chỉ phát hiện và nhận diện 1 lúc 1 biển báo. |

* Thuật toán này cho phép ngay lập tức phát hiện và nhận diện biển báo nhưng vì đòi hỏi lượng tính toán lớn và không phù hợp với mục đích của nhóm đề ra nên đã thay đổi và cải tiến phương pháp này.
* Do đó nhóm quyết định thử nghiệm phát hiện loại biển báo thông qua hình dáng và màu sắc, sử dụng phương pháp tìm viền của biển báo kết hợp với việc so sánh màu sắc giữa các vùng ảnh với nhau, nếu vùng được kiểm tra đáp ứng các nhu cầu về màu sắc, hình dáng và độ lớn thì sẽ tiến hành nhận diện biển báo.



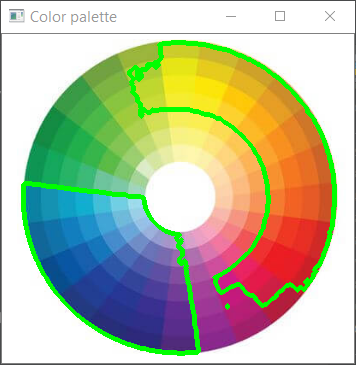
Hình 7. Sơ đồ thuật toán Phát hiện biển báo

* Các bước thực hiện phát hiện biển báo như sau:
* B1: Lọc ảnh với filter (3x3) sau đó chuyển đổi sáng hệ màu HSV để tối ưu cho việc so sánh độ sáng tối của ảnh. Thông thường ảnh input đầu vào sẽ ở dạng sắc nét, các đường bao của nhiều vật thể sẽ trồng lấn lên nhau, khiến cho việc tìm contour ở các bước sau khó khăn hơn. Đó là lí do nhóm chọn hàm GaussianBlur của thư viện OpenCV để làm mờ ảnh, giúp giảm độ sắc nét của các cạnh viền -> giảm số contour tìm được sau này. Thực tế nhóm có thử dùng các hàm làm mờ ảnh khác của python như hàm Blur và cũng đạt được kết quả tương tự. Nhóm dùng filter 3x3 vì qua kiểm tra thấy cho ra ảnh mờ ở mức độ phù hợp với yêu cầu bài toán.

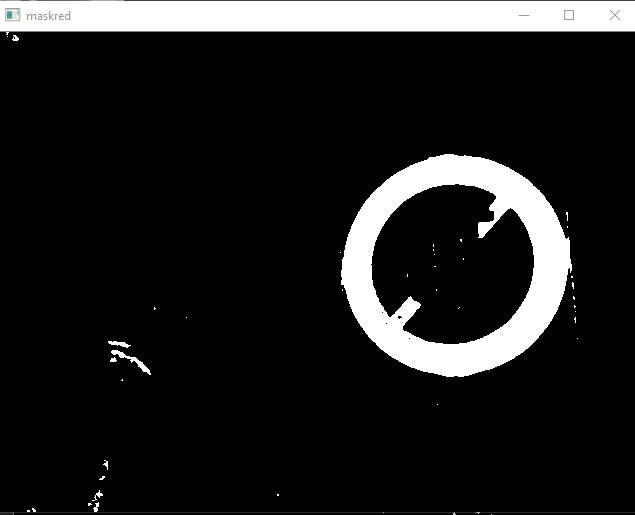
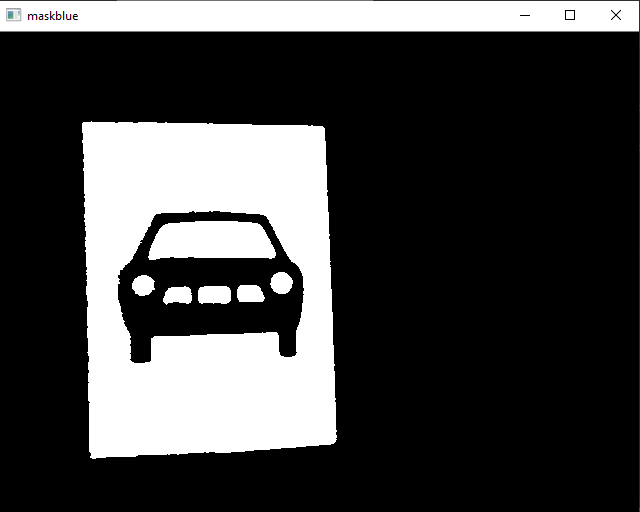


Hình 8. Mô tả hệ màu HSV

* B2: Tạo maskred và maskblue để lọc ra màu đỏ và xanh. Chọn các thông số phù hợp cho hàm inRange() để lấy được từng vùng màu.



Hình 9. Mô tả khả năng lọc màu của mask

Hình 10. Hình ảnh thực tế của các mask

* B3: Tìm viền to nhất theo mask từ B2 với điều kiện: diện tích viền lớn hơn 1 số nhất định (để tránh tình trạng bị nhiễu bởi một số vật thể nhỏ có màu sắc giống biển báo) và phải là 1 hình dạng có nghĩa (hình chữ nhật, tròn, tam giác). Lọc hình dạng của viền bằng cách xác định các góc của viền, sau đó dựa vào các góc của viền để xác định hình dạng.
* B4: Hiển thị phần hình ảnh đã được lọc qua để đưa vào bước Nhận diện.



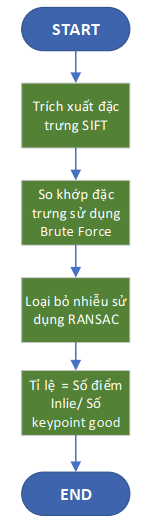
Hình 11. Hình ảnh biển báo sau khi đã được Phát hiện

* Ưu điểm:
* Có thể phát hiện biển báo mà không cần quan tâm về điều kiện sáng của ảnh.
* Phân biệt được nhiều loại và màu sắc của biển báo khác nhau.
* Nhược điểm:
* Sẽ bị nhiễu nếu có dị vật cùng màu đính sát biển báo.
* Phải cần thời gian để tối ưu cho mask phân biệt chính xác màu cần muốn.

# IV. Nhận dạng biển báo

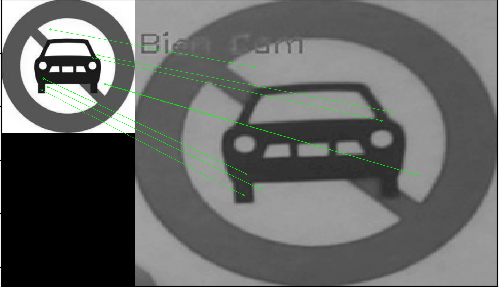
* Để Nhận dạng biển báo thì nhóm sử dụng 1 bộ thư viện biển báo ngoài (gồm 12 biển báo như bảng dưới) để so sánh với hình ảnh chiết xuất ra từ phần Phát hiện, bộ biển báo này được chia thành các nhóm khác nhau, mỗi nhóm có 1 đặc trưng riêng để xác định.

|  |  |
| --- | --- |
| Biển Cấm | Cấm người đi bộ Cấm xe oto Tốc độ tối đa 40 |
| Biển Nguy Hiểm | Đường bị hẹp 2 bên Giao nhau với đường k ưu tiên Vách núi nguy hiểm |
| Biển Chỉ Dẫn | Chỗ quay đầu Ưu tiên qua đường hẹp Đường dành cho oto |
| Biển Hiệu Lệnh | Chỉ được đi thẳng Đường cho người đi bộ Nơi giao nhau theo vòng xuyến |

* Mặc dù đặc trưng SIFT có tốc độ xử lý chậm, trích xuất ít đặc hơn ORB và SURF nhưng bù lại ở thực nghiệm thì SIFT có hiệu quả tính toán tốt hơn, đưa ra được tỉ lệ chính xác lớn hơn. Ngoài ra SIFT có khả năng phát hiện chính xác nhất cho các biến thể tỷ lệ, xoay và affine.

Hình 12. Lưu đồ giải thuật Nhận dạng

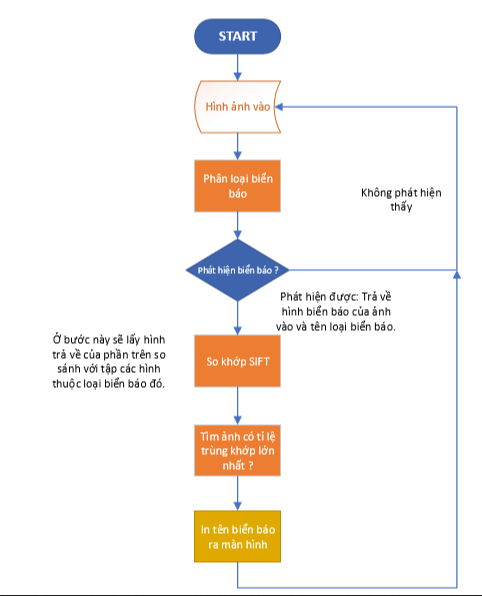
* Từ hình ảnh xuất ra từ phần phát hiện ta có thể dễ dàng xếp vào các nhóm biển báo để rút trích và so sánh đặt trưng SIFT.
* Các bước thực hiện nhận diện biển báo như sau:
* B1: Khởi tạo phần SIFT detector, tìm keypoints và descriptors của ảnh.
* B2: Từ descriptors chiết xuất từ B1 ta tiến hành so khớp 2 ảnh với nhau (ảnh từ phần phát hiện và ảnh từ thư viện biển báo). Sau đó ta sử dụng tỉ lệ khoảng cách thử nghiệm của Lowe’s để lưu trữ lại những điểm trùng khớp tốt.
* B3: Sử dụng hàm findHomography lọc lại những điểm trùng khớp chưa tốt.
* B4: Sau khi lọc ta tính ra số điểm Inlier từ đó tính ra được tỉ lệ giống nhau giữa 2 ảnh.



Hình 13. Trích xuất đặc điểm từ hình mẫu và hình ảnh từ phần Phát hiện

* Ưu điểm:
* Dễ dàng bổ xung hoặc thay đổi bất kì biển báo mà không cần thay đổi trong giải thuật.
* Nhận diện được chính xác biển báo bị nghiêng hay xoay góc khác.
* Nhược điểm:
* Thời gian xử lý chậm hơn một vài phương pháp khác (Template Matching, …).
* Cần hình ảnh mẫu đủ tốt để so sánh.

# V. Phương pháp kết hợp các thành phần

* Sau khi có được các thuật toán cần thiết cho chương trình, nhóm tiến hành kết hợp các thuật toán với nhau theo trình tự là: Xử lý video -> Phát hiện -> Nhận dạng. Các quá trình đòi hỏi phải kết hợp với nhau 1 cách tối ưu và hiệu quả, dưới đây nhóm sẽ giải thích rõ ràng cách thực hiện chương trình.
* B1: Lấy Frame từ video.
* B2: Sau khi có được Frame từ video ta tiến hành kiểm tra xem chương trình có cho phép Phát hiện biển báo hay không, nếu có ta tiến hành Phát hiện biển báo như phần trên đã mô tả, nếu không ta quay lại B1.
* B3: Sau khi thực hiện Phát hiện biển báo ta tiến hành Nhận dạng biển báo, nếu nhận dạng chính xác thì ta tiến hành bước tiếp theo, nếu không đạt yêu cầu thì ta quay lại B1.

Hình 14: Lưu đồ giải thuật chung của bài toán



Hình 15. Kết quả sau khi Phát hiện và Nhận diện

# VI. Tổng kết:

1. **Tự đánh giá và kết quả thực hiện:**

* Đồ án phát hiện, nhận diện và theo dõi biển báo giao thông Việt Nam được thực hiện trên ngôn ngữ Python, sử dụng các thư viện hỗ trợ sẵn như Opencv, Numpy, …, với việc ứng dụng dễ dàng và nhanh chóng, đồ án đã đáp ứng được các nhu cầu cơ bản đề ra.
* Khả năng phát hiện biển báo là 78% (78/100) đạt mức khá.
* Khả năng nhận diện biển báo là 99% (99/100) đạt mức tốt.
* Thời gian để xử lý hết chương trình tùy thuộc vào Fps (Frame pes seconds) của phần cứng, trong thực nghiệm với Fps khoảng 30 tốc độ xử lý cả chương trình gần bằng 1/6 giây.

1. **Điểm hạn chế:**

* Khoảng màu chưa chính xác, nên gặp nhiều trường hợp Phát hiện chậm, Phát hiện sai.
* Hình ảnh video bị giật lag do thời gian xử lý 1 Frames còn chậm.
* Chưa Phát hiện được những biển báo có hình thù đặc biệt.

1. **Hướng phát triển:**

* Tuy khả năng xử lý chính xác đạt mức yêu cầu nhưng vẫn cần phải cải thiện về độ chính xác, sửa các lỗi phát sinh trong quá trình xử lý.
* Giảm tốc độ xử lý bằng các thuật toán tối ưu về thời gian nhưng vẫn bảo đảm độ chính xác.
* Khả năng Phát hiện không còn phụ thuộc vào màu sắc mà thay vào đó là các đặt trưng riêng mà biển báo khác với các vật thể khác.
* Cập nhật thêm tính năng mới như lưu trữ lại các biển báo đã được nhận diện vào bổ xung cho thư viện tổng.
* Khả năng xử lý cùng lúc nhiều biển báo trên ảnh và cho ra kết quả tương ứng.
* Tích hợp khả năng bám vật thể đã được nhận diện (Tracking) để giảm thời gian xử lí, tăng tốc chương trình.
* Link DEMO: <https://drive.google.com/file/d/1rUfwf8Diz4iiUWRhoTGIZzG_n6x3k0QX/view?usp=sharing>

# VII. Tài liệu tham khảo

* Bộ hình ảnh biển báo được sưu tầm từ Google và video lấy trực tiếp từ thực tế.
* Tài liệu tham khảo:
* Từ Slide bài giảng, Google, Youtube, Git, ...
* Tài liệu về Opencv: <https://docs.opencv.org/master/d6/d00/tutorial_py_root.html>

----END----